NR INDEKSU 353965
PL ISSN 0860 – 1674

7

Z MIKROKOMPUTEREM NA TY

Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH

NR 7(31) LIPIEC 1988 CENA 150ZŁ



GHOST CHASER

Wiwat konkurencja!

Wakacje, a Ty wziąłeś jednak do ręki „Bajtka”? Dobrze zrobiłeś Czytelniku. Gdy głowy nie zaprzątają klasówki, czy sesja egzaminacyjna lepiej wchłania się wiedzę komputerową. A przy tym, o ile mniejsze wyrzuty sumienia: nie robisz wszak tego zamiast wkuwać biologię, czy historię. Mam nadzieję jednak, że rzeczywiście nie robisz tego zamiast, np. pływania na żaglówce. Z uroków lata nie warto rezygnować nawet na rzecz komputera, jak dowodzi przykład naszego redakcyjnego kolegi Wojtka Zientary, który gdy czytasz te słowa, halsuje po morzu zamiast układać dla Ciebie kolejny program na „Atari”.

A propos wyrzutów sumienia. Ciekawe, czy nie czują ich dziś, po zakończeniu kolejnego roku szkolnego szefowie naszej oświaty. Zapowiedziane przez uchwalony przed laty program „Juniory” nie popłynęły jakoś szeroką rzeką do szkół. Czyja w tym wina – nie wiadomo. Jak zwykle trwa przerzucanie piłeczki. A tak naprawdę, ani Was, ani mnie nie powinno interesować kto co ma zrobić. Pracownie, do których powrócicie we wrześniu, powinny już być wyposażone w sprzęt programowy, nauczyciele gotowi w każdej chwili udostępnić Wam to wszystko, czym dysponuje szkoła. Może ktoś wreszcie zda sobie sprawę z tego, że wprawdzie informatyka jest dla entuzjastów, ale zasady obsługi komputera powinny być znane każdemu, tak jak nauka poruszania się po ulicach.

Zabawa z „Juniorem” jest tym śmieszniejsza, że w ciągu dwóch lat od jego powstania nie zrobiono nic dla otwarcia drogi do nowego komputera, jego następcy. Na świecie do edukacji zaczynają przenikać już takie maszyny jak 32-bitowy Archimedes oparty o procesory typu RISC. Czy nie będziemy w tej sytuacji uczyć się o maszynach parowych, gdy wokół latać już będą samoloty?

Zostawmy jednak szkołę. Nad nią „Bajtek” chce się „pomęczyć” w numerze 9 poświęconym w dużej mierze edukacji. Dla entuzjastów informatyki domowej są bowiem nieco bardziej pomyślne wieści. Podczas tegorocznych Międzynarodowych Targów Poznańskich doszło wreszcie do podpisania, zapowiadanego przez nas już dość dawno, kontraktu między Pewexem, a reprezentującą Commodore amerykańską firmą BCI. Tak popularnym na naszym rynku małym Atari przybył zatem nowy konkurent. Nam wypada się tylko z tego cieszyć – większy wybór zawsze oznacza zmianę na lepsze.

Cieszymy się również z tego, że pomimo dość niewielkiego rozmiaru pierwszego kontraktu – 1200 sztuk – Commodore nie chce sprzedawać swoich wyrobów bez całej profesjonalnej otoczki. Szykuje wiele niespodzianek. Każdy nabywca Commodore ma ponoć otrzymać wraz z komputerem książkę w języku polskim, a zajmującym się tym komputerem klubom udostępnione będzie całe oprogramowanie „public domain”. Być może także w „bajtkowych” konkursach na liście nagród obok Atari pojawią się i Commodore.

Kto zatem chce kupić komputer może dziś poważnie zastanowić się nad wyborem: 65XE czy C-64, ST, a może Amiga? W wyborze tym może Wam pomóc lektura „Bajtka – Tylko o Commodore” i przygotowywanego właśnie „Tylko o Atari-2”. Nie liczcie natomiast na to, że kiedykolwiek wprost odpowiemy na pytanie, który komputer jest lepszy: X, czy Y? Nie odpowiemy, choćby z tego powodu, że nie mamy w redakcji szans na uzgodnienie opinii między sobą.

Grzegorz Onichimowski

ZA MIESIĄC:

W kolejnym, 8 numerze „Bajtka” znajdziecie m.in.

- **o muzyce z komputera: rozmowę z Wojciechem Mannem, Atari — pianinem, programy na nutę Commodore**
- **opis Atari 1040 ST**
- **mini-pakiet graficzny na Amstrada**
- **Urban Upstart i The Secret Diary of Adrian Mole**
- **pojedziemy także razem pociągiem przyszłości i przeżyjemy wspólne rozkosze łamania głowy nad kolejnym konkursem matematycznym**



A oto triumfator naszego Konkursu Świątecznego — Paweł Dziewit z Kozienic. Jeśli i Ty chcesz powtórzyć jego sukces wróć do poprzedniego numeru „Bajtka” i rozwiąż Konkurs z Szaesnaściorem. Termin nadsyłania odpowiedzi jeszcze nie upłynął!

Bajtek

"BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH" ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21 -12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domański- -redaktor naczelny „Sztandaru Młodych”.

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM” — kierownik zespołu „Bajtka”), Roman Poznański (z-ca sekretarza redakcji „SM” — sekretarz zespołu „Bajtka”), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Gogolewski, Andrzej Kowalewski, Andrzej Podulka, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski.

Klany redagują: Commodore — Klaudiusz Dybowski,
Amstrad-Schneider — Jonasz Mayer
Spectrum — Marcin Przasnyski,
Atari — Wojciech Zientara, Sergiusz Wolicki.

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wółtańska

WYDAWCA: RSW „Prasa-Książka-Ruch” Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.
Cena 150 zł.
Skład techniką CRT-200, przygotowalnią offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA- -KSIĄŻKA-RUCH” w Ciechanowie, ul. Sienkiewicza 51.
Nr zlecenia 010188 n. 150.000 egz.
U-113



SYSTEM SKRACANIA KOLEJEK

Rozmowa z Markiem Bisztyga, zastępcą informatyka wojewódzkiego w Urzędzie m.st. Warszawy.

— Pomysł, by za pomocą komputera tworzyć banki danych o ludności państw czy miast jest chyba lak stary jak i same komputery. Kiedy pojawił się w naszym kraju?

— Nie wiem kto i kiedy wpadł na ten pomysł. Z całą pewnością można natomiast powiedzieć, że prace nad systemem „PESEL" (Powszechny Elektroniczny System Ewidencji Ludności) rozpoczęły się w Polsce w 1988 roku. Początkowo prowadzono tylko prace eksperymentalne. Obecnie wyszliśmy już dawno z fazy eksperymentu. W stolicy wprowadziliśmy do systemu, nazwanego przez nas Terenowym Bankiem Danych i będącego częścią PESEL-a, dane o wszystkich mieszkańcach Warszawy, a obecnie „wpisujemy" tam mieszkańców województwa stołecznego.

— Jak zorganizowana jest tak wielka baza danych ? Przykładowo, na jakim pracujecie sprzęcie?

— Jednostką centralną systemu jest dość już zaawansowany w latach dzierżawiony przez nas z Rządowego Centrum Informatycznego komputer marki „Siemens". Technologia obsługi systemu też nie jest najnowocześniejsza. W poszczególnych urzędach i dzielnicowych, a lakże w innych współpracujących z nami instytucjach znajdują się nieinteligentne terminale. Można dzięki nirn wprawdzie szybko sięgać po informacje, bądź też je aktualizować, lecz, z drugiej strony, taka organizacja naszej bazy mocno ogra-.czona ilość możliwych aplikacji.

— Czyli każdy użytkownik ma tylko ściśle określony zasób informacji?

— Tak i to w dodatku niezbyt szeroki. Próbowaliśmy zmieniać tę technologię poprzez wprowadzenie do naszej bazy pewnych podsystemów tematycznych. Baza przechowuje rekordy (jeden odpowiada jednemu mieszkańcowi) o jednolitej strukturze, różnicuje się natomiast dla poszczególnych użytkowników moduły aplikacyjne. I tak biuro meldunkowe, wydział ruchu drogowego czy wydział komunikacji korzystają częściowo z tych samych danych — np. imię, nazwisko, miejsce zamieszkania, częściowo zaś wykorzystują pewne dane z poszczególnego rekordu zastrzeżone wyłącznie dla nich.

— Taka organizacja bazy powoduje jednak, że ciągle zwiększa się długość każdego pojedynczego rekordu?

— Dlatego też rezygnujemy już w zasadzie z poszerzenia tym sposobem ilości informac|i tematycznych. Chcemy zmienić nieco „filozofię" naszego systemu w kierunku tworzenia rozproszonej bazy danych. Nie będzie to wprawdzie jeszcze typowa sieć otwarta, ale centralny bank danych uzupełniony będzie w niej przez szereg banków lokalnych. Zamiast zwy kłego terminalu w każdym współpracującym z nami urzędzie zainstalowany będzie mikro-lub minikomputer. Bank centralny, w tej sytuacji, zawierać będzie tylko podstawowe dane identyfikacyjne natomiast cała ich reszta zapisana będzie w sieci mikrokomputerów.

— Czy nie przewidujecie kłopotów z łącznością? Wszak nasze łącza telefoniczne nie należą, oględnie mówiąc, do najnowocześniejszych.

— Gdybyśmy mieli czekać na powstanie zapowiedzianej ogólnopolskiej sieci komputerowej z systemu mogtyby korzystać chyba dopiero nasze dzieci. Tak poważnie jednak, nie mamy zbyt wielkich problemów z łączami. Transmitować nimi będziemy tylko to, co konieczne. Zastosujemy opracowane specjalnie dla nas modemy synchroniczne.

— Pomówmy przez chwilę, o innym nieco, nie związanym ściśle z techniką aspekcie Waszej pracy. Czy nie spotykacie się z zarzutem, że tego typu ewidencja danych jest zamachem administracji na wolność osobistą każdego z nas, pra- . wo do prywatności itd?

— Takie zarzuty pojawiały się wszędzie tam, gdzie wprowadzono podobne systemy komputerowe. Kojarzyły się one ludziom z wizją państwa policyjnego, które wie wszystko o każdym z obywateli i w każdej chwili może te informacje obrócić przeciwko niemu. Dyskusja na ten temat toczyła się m.in. w RFN wówczas, gdy wprowadzono tam nowy typ dowodów osobistych.

Nasz system nie ma nic wspólnego z tajnym dossier. W zasadzie niemal wszystkie informacje zawarte w każdym rekordzie podawane są przez samego obywatela — wówczas, gdy zgłasza on urodzenie dziecka, wstępuje w związek małżeński itp. Wyjątek od tej reguły stanowią tylko informacje Wydziału Ruchu Drogowego, który gromadzi dane niesfornych kierowców ukaranych mandatami.

— IV Warszawie nie należy zatem wyrzucać mandatu kredytowego?

— Nie radziłbym.

Wracając jednak do poprzedniego pytania. Otóż, jak się można łatwo domyśleć, niemal wszystkie dane zawarte w systemie są objęte tajemnicą. Sięgać po nie mogą tylko osoby do tego upoważnione.

— Jednak każde zabezpieczenie można złamać.

— Żaden system, także i nasz nie jest w 100 procentach bezpieczny Stosujemy oczywiście, różne zabezpieczenia, |ak np. system haseł otwierających „furtki" do kolejnych informacji.

Najważniejsze jest jednak to, że za wyjątkiem wspomnianego banku danych o piratach drogowych, a także prowadzonej przez administrację ewidencji podatków czy gruntów, nie widzę specjalnego sensu „włamywania" się do systemu, którego większość danych zdobyć można znacznie łatwiej całkiem legalną drogą.

— Jeśli systemu nie może użyć potencjalny „hacker" do czego poza wprowadzaniem statystyk potrzebny jest on administracji?

— Najważniejsza iest właśnie wspomniana ewidencja, zastąpienie tysięcy segregatorów papierowych znacznie łatwiejszą w użyciu informacją elektroniczną. W Warszawie, co należy szczególnie podkreślić system Terenowego Banku Danych działa już na równych prawach z ewidencją papierową. Gdy, na przykład, obywatel zgłosi się do urzędu celem zameldowania się meldunek staje się faktem w momencie zapisania go na terminalu komputera. Żadne dodatkowe czynności biurowe nie są już potrzebne. Z danych meldunkowych korzysta wiele instytucji np. PZU. Poza tym TBD wykorzystywany jest do innych celów. Dla przykładu listy wyborców podczas ostatnich wyborów do rad narodowych sporządzane były w stolicy wyłącznie w oparciu o dane z TBD.

— Wróćmy do informatycznej strony zagadnienia. Czym różni się Wasz system od np. dBase III?

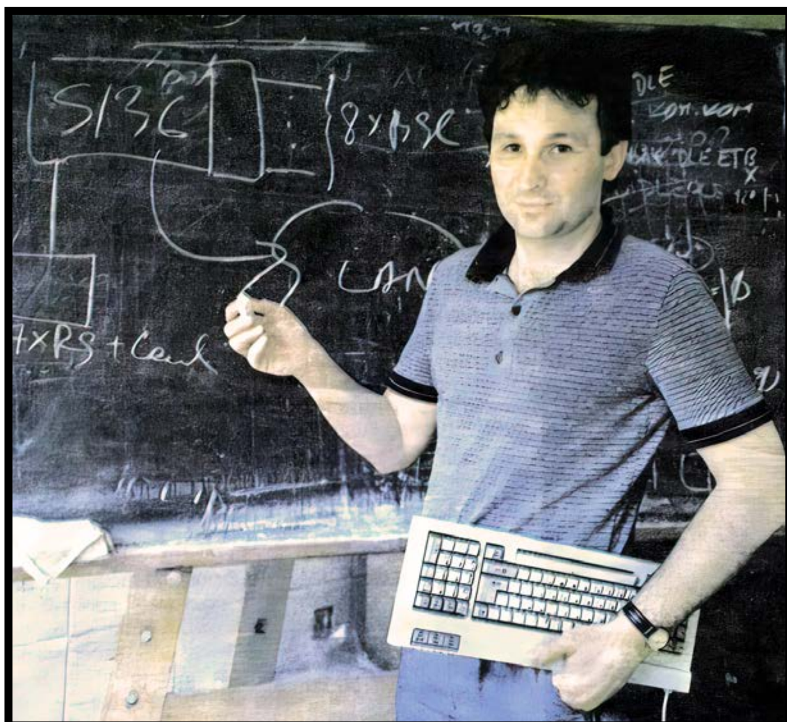
— Jeśli chodzi o zasadę działania, to niemal niczym. Każda baza danych, bowiem, to zbiór danych z aplikacjami. Natomiast z tajemnic „kuchni" mogę zdradzić to, że pierwsza wersja systemu oparta była na oprogramowaniu firmowym dostarczonym wraz z komputerem. Po latach stworzono praktycznie nowy system, od początku do końca opracowany tu w Warszawie. Braliśmy pod uwagę możliwość posłużenia się jednym z istniejących systemów, np. dBase, lecz zarzuciliśmy szybko ten pomysł. Istniejące oprogramowanie narzędziowe ma sporo błędów do których źródeł dotrzeć jest równie trudno, jak opracować własny software. Nasz system musi być niezawodny. Lepiej zatem wiedzieć dokładnie, co się ma w ręku, odpowiadać za to w pełni, mieć możliwość ciągłego nadzoru, wprowadzania zmian itp.

— Czy wierzy Pan jednak, w to, że dzięki Waszemu systemowi znikną kolejki w urzędach?

— Z pewnością przynajmniej część się zmniejszy. Czy zresztą nie widział Pan, co dzieje się choćby w wydziałach komunikacji? O ileż łatwiej byłoby chociażby zarejestrować samochód, gdyby urzędnicy mieli wciąż pod ręką a raczej „pod klawiaturą" wszystkie potrzebne dane o pojazdach i ich użytkownikach?

Można zatem obruszać się na nadawanie obywatelom numerów ewidencyjnych, gromadzenie informacji o nich. Te informacje jednak już wcześniej były przecież w rękach administracji. Niestety cały szkopuł w tym,' że czasem wykorzystywano je przeciw obywatelowi. Założeniem naszego systemu jest, by tak łatwy i wielostronny dostęp do danych służył każdemu z nas. Rzeczywiste efekty jednak to najczęściej ani nasza wina ani zastuga. Odpowiedź na to pytanie; na ile system nasz zmniejszy biurokrację, złagodzi dolę petenta, nie należy do informatyków. My dajemy tylko narzędzie. Jesteśmy przekonani o tym, że będzie ono narzędziem sprawnym.

Rozmawiał Grzegorz Onichimowski





DRUKARKA

CITIZEN MSP-15E

Citizen MSP-15E jest 9-igłowa drukarką mozaikową z 15-calowym wałkiem do komputerów osobistych (domowych i biurowych). Drukarka ta jest modyfikacją poprzedniego modelu MSP-15, a ten z kolei jest rozwinięciem testowanej w „Bajtku" 4/88 drukarki Citizen 120-D. Większość informacji tam podanych odnosi się także do tej drukarki. Zwrócę więc jedynie uwagę na występujące różnice.

BUDOWA I DZIAŁANIE

Na pierwszy rzut oka drukarka wygląda dość delikatnie, szczególnie w porównaniu z NX-15 (Star). Wrażenie to jest jednak bardzo mylące, a powoduje je „lekkość" kształtów. Sama drukarka ma solidną, klasyczną konstrukcję mechaniczną.

Prędkość druku została zwiększona do 160 CPS (Character Per Second — znaków na sekundę) w trybie dratt orraz do 40 CPS w trybie korespondencyjnym (NLQ). Zestaw krojów pisma pozostał niezmieniony i zawiera wszystkie standardowe kroje: Pica — 10 CPI (Character Per Inch — znaków na cal), Elitę — 12 CPI, Condensed Pica — 17 CPI oraz Condensed Elitę (20 CPI). Przy większej długości wałka daje to odpowiednio 136, 163, 231 i 272 znaki w wierszu.

Zachowane zostały także dodatkowe odmiany pisma: poszerzony (odpowiednio 5, 6, 8,5 i 10 CPI), pochylony, odwrócony, proporcjonalny i o podwójnej wysokości. Razem z powszechnymi odmianami druku wytłuszczonego (double strike) i zagęszczonego (emphasized) daie to bardzo szeroką gamę możliwości — większość krojów i odmian można ze sobą złączyć.

Standardem Citizena są już odstępy wierszy programowane w trzech podziałkach: n/72, n/144 i n/216. MSP-15E posiada równie bogate jak 120-D możliwości tabulacji, justowania oraz innych sposobów kształtowania wydruku. Zainteresowanych odsyłam do opisu drukarki Citizen 120-D.

EKSPLOATACJA

Testowana drukarka MSP-15E pracowała przez dwa miesiące w redakcji, jako podstawowy sprzęt. Przez cały ten okres spisywała się bardzo dobrze, zarówno przy współpracy z komputerami 8-bitowymi (Atari, Am- strad, Commodore), jak i z komputerem Amstrad PC 1640 (zgodny z IBM PC). Dał się zauważyć jedynie nieco większy poziom hałasu niż w innych drukarkach tej firmy. Jednakże trudno jest ocenić, czy jest to wada całej serii, czy tylko testowanego egzemplarza.

Praca z drukarką jest wygodna. Dzięki dużej liczbie kodów sterujących można programowo uzyskać wszystkie dostępne efekty. Z tego powodu praktycznie nie była wykorzystywana możliwość zmiany trybu pracy drukarki przy pomocy jej klawiszy sterujących. Kłopotliwa okazała się tylko zmiana parametrów pracy przy pomocy mikroprzełączników. Są one umieszczone wewnątrz drukarki i dokonanie zmiany wymaga zdjęcia obudowy. Uważam to za krok wstecz w konstrukcji tego udanego urządzenia.

PODSUMOWANIE

Drukarka Citizen MSP-15E posiada możliwości wydruku tekstu i grafiki, które pozwalają zaliczyć ją do sprzętu prawie profesjonalnego. Szeroki wałek, długa taśma oraz duża trwałość głowicy drukującej predestynują ją przede wszystkim do zastosowań biurowych. Może być także przydatna dla użytkowników indywidualnych wykonujących często duże ilości wydruków. Przy drukowaniu na pojedynczych arkuszach daje się jednak odczuć brak podajnika, który jest już standardowym wyposażeniem drukarek innych firm. Dystrybutor tego sprzętu, firma Synelec i serwisujące go „Ipaco" pobiły natomiast wszytkich na innym polu — jest to pierwsza drukarka jaką otrzymaliśmy z instrukcją napisaną PO POLSKU.

Zalety drukarki:

— instrukcja w języku polskim
— duża liczba zestawów znaków w trybie Epson
— dodatkowe znaki graficzne w trybie Epson
— duża liczba krojów pisma i ich kombinacji
— programowe przełączanie trybów Epson IBM
— duża trwałość głowicy drukującej
— duża kaseta z taśmą, która zapewnia długotrwałe użytkowanie bez regeneracji

Wady drukarki:

— umieszczenie przełączników trybów pracy wewnątrz obudowy
— brak podajnika papieru
— brak krawędzi do odrywania papieru z rolki
— stosunkowo głośna praca.

Wojciech Zientara

PARAMETRY TECHNICZNE DRUKARKI CITIZEN MSP-15E WG SPECYFIKACJI PRODUCENTA:

głowica	— 9-igłowa
szybkość druku	— 160 CPS w trybie draft, 40 CPS w trybie NLQ
matryca znaków	— 9 na 9 punktów (draft). 17 na 17 punktów (MLQ)
wielkość znaków	— 2,4 na 2,4 mm
gęstość druku	— 20, 17, 15, 12, 10, 8.5, 6 i 7 CPI
gęstość grafiki	— 60, 72, 80, 90, 120, 144 i 240 DPI
odstęp wierszy	— 1/6, 1/8, 7/72, n/72, n/144 i n/216 cala
papier	— szerokość 5-16 cali grubość 0,06-0,3 mm
trwałość głowicy	— 100 min znaków
interfejs	— równoległy, 8- lub 7-bitowy Centronics (Amphenol)
zasilanie	— 180-264 V, 50-60 Hz, max. 90 VA
wymiary	— długość — 354 mm szerokość — 579 mm wysokość — 90 mm
masa	— 7 kg

U.S.A.
Francja
Niemcy
Anglia
Dania I
Szwecja
Wlochy
Hiszpania
Japonia
Norwegia
Dania II

Zestawy międzynarodowe

128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159

Dodatkowe znaki graficzne

TRYB 0
TRYB 1
TRYB 2
TRYB 3
TRYB 4
TRYB 5
TRYB 6
TRYB 7

Tryby graficzne

SYNFILE+

Z pewnością najpopularniejszą i najlepszą bazą danych dla komputerów Atari XL/XE jest SynFile+ znanej amerykańskiej firmy Synapsę Software.

Program ten wymaga komputera z pamięcią 64 lub 128 KB oraz jednej lub dwóch stacji dysków. Umożliwia założenie zbioru danych o praktycznie dowolnej pojemności. Dla każdego zbioru można zdefiniować dowolny format rekordu, w którym będą zapisywane informacje. Maksymalna wielkość rekordu jest ograniczona tylko rozmiarem ekranu: 21 wierszy po 80 (tak, osiemdziesiąt!) znaków. Oczywiście rekord może być dowolnie podzielony na pola różnych typów. Do tworzenia nowego zbioru służy specjalny proram „Create", który znajduje się na dyskietce SynFile'a.

Rodzaje pól

Podczas tworzenia struktury zbioru danych użytkownik ustala położenie, nazwę, wielkość i rodzaj elementów struktury zwanych polami. W jednym rekordzie można umieścić do 66 pól. Po ustawieniu kursora na wybranym miejscu należy wpisać nazwę pola. Nazwa może składać się z dużych i małych liter, cyfr i innych symboli (oprócz nawiasów). Maksymalna długość nazwy wynosi 31 znaków. Naciśnięcie **RETURN** zatwierdza podaną nazwę i powoduje wyświetlenie dodatkowego menu, które zawiera dozwolone rodzaje pól. Są one następujące:

Text — pole tekstowe, które może zawierać dowolne znaki (litery, cyfry, symbole, a także znaki pseudograficzne — z CONTROL). Maksymalna długość pola tekstowego wynosi 255 znaków.

Look-Up - pole tablicowane, którego zawartość musi znajdować się w tabeli. Po zakończeniu definiowania rekordu program prosi o wprowadzenie do tabeli wszystkich dopuszczalnych elementów. Przy zapisywaniu danych w polu Look-Up mogą być wpisane tylko takie elementy, które zostały wcześniej zdefiniowane (np. nazwy województw). Pola Look-Up pozwalają na zaoszczędzenie nośnika, gdyż na dyskietce zapisywany jest tylko numer elementu. Długość pola jest za to ograniczona do 34 znaków.

Date — pole daty, które zawiera datę w formacie amerykańskim — miesiąc /dzień/rok. Pole ma zawsze 8 znaków długości, a znaki „/" są wpisywane automatycznie. Program jest zabezpieczony przed wprowadzeniem niewłaściwej daty (np. miesiąc 15).

Numeric — pole liczbowe pozwalające na wprowadzenie dowolnej wartości z zakresu od -9x 1097 do 9x1097. Po wybraniu tego typu pola program pyta, czy położenie przecinka ma być ustalone (fixed), a jeśli tak, to ile będzie cyfr do przecinku

Integer — pole liczb całkowitych. Może zawierać liczby całkowite z zakresu od -32768 do 32767. Warto go używać wszędzie tam, gdzie występują tylko wartości całkowite, gdyż pole **Numeric** zajmuje sześć bajtów, a **Integer** tylko dwa.

Dollar — pole wartości pieniężnej, w którym każda liczba jest poprzedzana znakiem „$" (program powstał w USA) oraz ma dwie cyfry po orzecinku. Przy określaniu długości tego pola trzeba uwzględnić miejsce na punkt dziesiętny (przecinek), ale nie na „$", gdyż ten znak jest umieszczony przed polem.

Computed — pole liczbowe o wartości obliczanej na podstawie zawartości innych pól według podanej formuły. Formuła jest wprowadzana podczas definiowania rekordu (Create). Obliczenie jest wykonywane automatycznie po wpisaniu rekordu.

Cumulative — pole liczbowe zawierające sumę innych pól i jego samego. Zawartość tego pola również jest obliczana automatycznie.

Conditional — pole warunkowe. Przy definiowaniu rekordu ustalane są dwa wpisy (typu **Text**, np. ZYSK i STRATA) oraz formuła warunku. Jeżeli obliczenie formuły warunku da wynik dodatni, to w polu umieszczany jest pierwszy wpis (ZYSK), a jeśli ujemny, to drugi (STRATA).

Record — numer wprowadzonego rekordu. Komputer wpisuje tu automatycznie kolejny numer licząc od 1. Maksymalna wartość w tym polu może wynosić 32767. Jest to najwygodniejszy sposób numerowania wprowadzanych danych, lecz numeracja |est wykonywana według kolejności wpisywania i nie można jej zmienić. W rekordzie może być tylko jedno pole typu Record.

Counter — pole licznika. Jego działanie jest podobne do Record. ale użytkownik określa liczbę początkową (od 0 do 999) i wartość, o jaką zwiększany jest licznik przy każdym nowym wpisie (od 1 do 100). Pole typu **Counter** może być także tylko jedno w rekordzie.

Po ustaleniu rodzaju pola trzeba jeszcze określić jego długość (oprócz **Date**) oraz sposób justowania wpisanej danej (do lewej lub prawej krawędzi pola). Jeżeli do pola liczbowego nie zostanie wpisana żadna informacja, to program samoczynnie umieszcza w nim zero. Pola typu **Record, Counter, Computed** i **Conditional** są wypełniane przez program.

Stworzoną strukturę rekordu można zmienić przez wybranie z programu „Create" funkcji „Edit file> (Redagowanie). Poza dodaniem i usunięciem pól możliwe jest również dokonanie w nich zmian. Program udostępnia możliwości: przesunięcia pola, zmiany nazwy, zmiany typu i zmiany długości.

Operacje na zbiorach

Często podczas pracy ze zbiorami danych występuje konieczność wykonania operacji na całości lub części zbioru (np. skopiowanie lub podzielenie). Dla umożliwienia tych i innych operacji **SynFile+** posiada odpowiednie funkcje dostępne po wybraniu opcji **FILES**.

Open — otwarcie zbioru danych. Po wywołaniu tej funkcji wyświetlany jest spis zbiorów umieszczonych na dyskietce znajdującej się w stacji dysków. Po wybraniu kursorem odpowiedniego zbioru program odczytuje z niego niezbędne informacje, a następnie czeka na dalsze polecenia.

Close — zamknięcie zbioru danych. Wykonanie tej operacji jest konieczne po zakończeniu pracy z danym zbiorem, jeżeli zostały dokonane jakieś zmiany w jego zawartości. W przeciwnym razie dokonane zmiany mogą nie zostać zapisane na dyskietce. Gdy po dokonaniu zmian wybrana zostanie funkcja **Open**, program sam wykonuje najpierw funkcię **Close**.

Copy — kopiowanie zbioru danych. Umożliwia przeniesienie całego zbioru na inną dyskietkę lub wykonanie kopii na tej samej dyskietce. W celu uniknięcia pomyłek podczas kopiowania przy użyciu jednej stacji dysków zbiory źródłowy i docelowy muszą mieć różne nazwy (program tego nie sprawdza!).

Subfile — tworzenie podzbioru. Po wybraniu tej funkcji i podaniu nazwy nowego zbioru, program pyta o kryteria kopiowania poszczególnych rekordów do nowego zbioru. Obowiązują tu te same ograniczenia, co w funkcji **Copy**.

Merge — łączenie zbiorów. Ta funkcja pozwala na dołączenie jednego zbioru do drugiego. Zawartości pól ze zbioru źródłowego są przepisywane do zbioru docelowego do pól o tej samej nazwie i zbliżonym typie. W tym miejscu autorzy programu nie popisali się. Rekordy są bowiem przenoszone po jednym i przy użyciu jednej stacji i dwóch dyskietek wymaga to wymiany dyskietek tyle razy, ile rekordów jest przenoszonych. Trzeba więc korzystać z dwóch stacji dysków, albo wykonywać tę operację w obrębie jednej dyskietki.

Rename — zmiana nazwy zbioru. Funkcja ta pozwala na zmianę nazwy dowolnego zbioru, a działa podobnie jak w DOS-ie. Ponieważ jednak każdy zbiór składa się conajmniej z trzech plików, to użycie do tego celu DOS-u jest bardziej kłopotliwe.

Delete — usunięcie zbioru z dyskietki — ta operacja nie wymaga żadnego komentarza.

Density — zmiana gęstości zapisu na dyskietce. SynFile+ rozróżnia trzy gęstości: pojedynczą (single), podwójną (double) i rozszerzoną (1050). Ustawienie gęstości musi być wykonane przed formatowaniem nowej dyskietki danych. Pozniei- sze zmiany gęstości są już niemożliwe. Nie można także kopiować i przenosić zbiorów między dyskietkami zapisanymi w pojedynczej i podwójnej gęstości.

Format — formatowanie dyskietki przeznaczonej do zapisu zbiorów danych. Dyskietki przeznaczone na zbiory danych **SynFile+** należy zawsze formatować przy jego pomocy, ponieważ format **SynFile+** różni się nieco od standardowego (szczególnie w gęstości 1050).

SynFile → DIF — zmiana formatu zapisanych danych. Aby umożliwić przenoszenie danych pomiędzy różnymi programami opracowany został standardowy format zapisu informacji na dyskietce DIF (Data Interchange Format). Zbiory tworzone przez **SynFile+** nie odpowiadają jednak temu formatowi przed użyciem ich np. w **SynStat** czy **SynGraph**. trzeba zmienić ich format.

DIF → SynFile — odwrotna operacja zmiany formatu z DIF na **SynFile+** umożliwia dołączenie do zbiorów SynFile'a danych utworzonych przy pomocy innych programów.

Podczas przeprowadzania wymienionych wyżej operacji żądania komputera i konieczne informacje są wyświetlane w oknie menu. Ponieważ program zawiera kilka błędów warto przed przystąpieniem do łączenia, dzielenia i kopiowania zbiorów wykonać najpierw kopie całej dyskietki.

Operacje w zbiorze

Podstawowymi czynnościami, do których przeznaczona jest baza danych, nie są jednak operacje na zbiorach danych, lecz operacje na danych. Ich< przeprowadzenie jest możliwe po otwarciu zbioru.' przez wybranie opcji **RECORDS**

Enter — wprowadzanie danych. Zdefiniowanie zbioru to jeszcze nie wszystko. Otrzymujemy w ten sposób jedynie tabelki, które trzeba wypełnić treścią. Umożliwia to funkcja **Enter**, która powoduje wyświetlenie pustego rekordu. Po wpisaniu odpowiedniej treści, naciśnięcie klawisza **START** umieszcza zawartość rekordu w zbiorze. Wprowadzanie danych jest przerywane przez naciśnięcie **SELECT**

Retrieve — przeglądanie danych. Przy pomocy tej funkcji można przeglądać i poprawiać wszystkie dane znajdujące się w zbiorze. Po jej wybraniu należy podać kryteria wyszukiwania informacji w zbiorze lub nacisnąć **START**, gdy chcemy przeglądać wszystkie rekordy. Po wyświetleniu rekordu można wykonać w nim dowolne zmiany lub naciśnięciem klawisza **OPTION** wywołać dodatkowe menu. Umożliwia ono wydrukowanie lub usunięcie rekordu, a ponadto przeliczenie zawartości jego pól liczbowych.

Re-index — porządkowanie zbioru. **SynFile+** wymaga, aby zbiór danych był uporządkowany. Dlatego też operacja ta jest wywoływana automatycznie przy pierwszym otwarciu zbioru nowoutworzonego lub powstałego z przyłączania innego zbioru. Ponadto można ją wywołać w każdej chwili w celu zmiany uporządkowania zbioru. Najpierw wyświetlany jest wykaz nazw pól, z którego należy wybrać pole (lub pola), według którego będą porządkowane dane. W przypadku pól typu Text trzeba jeszcze podać, ile początkowych znaków z tego pola ma być uwzględnione. Teraz w górnej linii ekranu pojawia się informacja o liczbie znaków użytych do porządkowania zbioru oraz aktualna i maksymalna liczba rekordów, które mogą być w ten sposób uporządkowane. Wybór pól indeksowych kończy się po ustawieniu kursora na słowie **DONE**. Następnie program pyta jeszcze, czy kolejność porządkowania ma być rosnąca, czy malejąca i przystępuje do sortowania zbioru. Na koniec sposób uporządkowania jest zapisywany na dyskietce.

Delete all — kasowanie. Jeżeli zachodzi potrzeba usunięcia ze zbioru większej liczby rekordów, to wykonywanie tego pojedynczo jest bardzo żmudne. Należy wtedy użyć funkcji Delete all. Komputer najpierw pyta o kryteria usuwania rekordów, a po ich

podaniu i potwierdzeniu wykonuje żądaną operację. Każdy usuwany rekord jest przy tym wyświetlany na ekranie, co umożliwia w pewnym stopniu kontrolowanie operacji.

Update all — zmiana zawartości. Podobne działanie ma funkcja zmiany zawartości rekordów. Jednak zamiast kasowania zamienia ona wskazany element rekordu na inny, który również musi być podany. Ponieważ żaden rekord nie może być powielony (skopiowany na inny rekord), to funkcja **Update all** umożliwia ominięcie tej niedogodności. W rekordach, w których zawartość jednego lub kilku pól ma być bardzo podobna, wpisujemy dane skrótowo (najlepiej przy pomocy nietypowych znaków), a następnie przy pomocy **Update all** zamieniamy ich zawartość na poprawną. Ewentualne drobne korekty można potem wykonać indywidualnie w każdym rekordzie.

Warto wiedzieć, że po żądaniu przez program podania kryteriów wyszukiwania nie trzeba wpisywać całej zawartości pola. Można użyć znaku *" na podobnych zasadach, jak podczas pracy z DOS-em, lecz zarówno za, jak i przed szukanym fragmentem pola (lub przed i za). Ponadto przy wyszukiwaniu danych można używać znaków „<" oraz „>" i to także w stosunku do pól tekstowych (litery są sortowane w kolejności: duże — małe, więc „a" jest za „Z", a nie między „A" i „B").

Raporty

SynFIle+ umożliwia także wykonywanie zbiorczych zestawień danych oraz tzw. etykiet. Rezultat tych operacji może zostać wyświetlony, wydrukowany lub zapisany na dyskietce. Funkcje te są dostępne po wybraniu opcji REPORTS.

Lists — listy. Po wybraniu tej funkcji w górnej części ekranu wyświetlane są nazwy pól, a w dolnej linia z liczbami. Linia ta określa liczbę znaków w jednym wierszu listy. Dla ekranu nie należy przekraczać wartości 40. Wartość dla zapisu na dyskietce jest dowolna, a dla wydruku zależy od trybu pracy drukarki. Po wpisaniu nazw pól w dolnej linii (nazwy muszą być oddzielone conajmniej jedną spacją) komputer rozsuwa je odpowiednio i pyta o urządzenie, na które ma wysłać listę. Po wybraniu stacji dysków pyta się jeszcze o nazwę pliku, a po wybraniu drukarki o tytuł wydruku. To ostatnie pytanie umożliwia podanie drukarce w tytule kodów sterujących i uzyskanie np. zwężonego lub wyrazistego druku. Na końcu należy jeszcze podać kryteria wyszukiwania rekordów ze zbioru.

Labels — etykiety. Ta funkcja w odróżnieniu od poprzedniej nie umieszcza danych z rekordu w jednej linii, lecz w polu o zdefiniowanym kształcie. Pożądany wygląd etykiety uzyskujemy przez wpisanie nazw pól w miejscach, w których ma się znaleźć ich zawartość. Maksymalny wymiar etykiety jest ograniczony rozmiarem ekranu (21 wierszy i 80 kolumn), a liczba pól w etykiecie nie może przekraczać 32. Po wybraniu urządzenia wyjściowego program pyta o parametry etykiet. Kolejno należy podać: liczbę etykiet obok siebie, lewy margines, poziomy (w spacjach) i pionowy (w liniach) odstęp między etykietami. Na zakończenie trzeba ustalić kryteria przeszukiwania zbioru.

Pomimo kilku błędów w programie oraz istnienia wielu innych baz danych dla komputerów Atari XL/ XE, SynFile+ wydaje się być bezkonkurencyjny w zakresie oferowanych możliwości. Jest to program wystarczający w zupełności do prowadzenia kartoteki, a w połączeniu z SynCalc także księgowości, w niewielkim zakładzie. Nawet serwis Atari w początkach działalności miał kartotekę prowadzoną przy pomocy SynFile+ — cały zbiór danych zajmował prawie dziesięć dyskietek.

Wojciech Zientara



ANIMACJA NA 130XE

Komputery Atari 130 XE ze względu na większą pamięć mają znaczną przewagę nad innymi ośmiobitowymi komputerami tej firmy. Jedną z ciekawszych możliwości jest animacja z wykorzystaniem pamięci dodatkowej. Pozwala ona na zastosowanie trybów graficznych o dużej rozdzielczości, np. trybu graficznego 8 + 16, tj. trybu 8 bez okna tekstowego. Obraz w GRAPHICS 8 + 16 zajmuje niemalże 32 strony pamięci RAM, dlatego w komputerach bez pamięci dodatkowej można umieścić jednocześnie co najwyżej cztery takie obrazy.

Przełączanie banków pamięci w Atari 130 XE uzyskuje się poprzez zmianę zawartości komórki pod adresem 54017. Blokowy schemat pamięci Atari 130 XE został podany w „Bajtku" 5/87, a sposób przełączania pamięci dodatkowej jest opisany w podręczniku załączanym do komputera, ale błędnie podana jest zawartość komórki 54017. Pod adresem tym, po włączeniu komputera znajduje się wartość 253, a nie 193 — jak podaje podręcznik, co oznacza, iż zarówno CPU tzn. procesor 6502C jak i ANTIC maja dostęp do pamięci podstawowej ("Normal"), a nie do dodatkowej.

Animacja opisana tutaj jest oparta na podobnej zasadzie, co przedstawiona w „ Bajtku" 5-6/86. Istotną jednak różnica jest to, że w przedstawionym programie następuje nie tylko przełączanie starszego bajtu adresu programu ANTIC-a, lecz także banków pamięci. Układając własny program należy zwrócić uwagę na to, aby tak rozmieścić obrazy, żeby ich przełączanie odbywało się za pomocą tylko jednej instrukcji POKE, tzn. albo tylko przełączenia banków, albo w wyniku zmiany adresu programu ĄNTIC-a. W przeciwnym wypadku nastąpi nakładanie się obrazów i ruch na ekranie nie będzie płynny.

Zamieszczony program przykładowy rysuje osiem faz ruchu graniastosłupa, a następnie przechodzi do przełączania kolejnych obrazów, co daje złudzenie obrotu bryły. Klawiszami „START" i „SELECT" reguluje się prędkość obrotu. Zatrzymanie programu następuje po naciśnięciu klawisza „BREAK" lub „RESET".

Po wprowadzeniu programu można na nim poeksperymentować poprzez wpisanie nowych zmiennych lub zmianę wartości zmiennych użytych w programie i uzyskać przez to graniastosłupy (a nawet inne bryły) o różnych wymiarach.

Linia 10	—
Linia 20	— Ustawienie zmiennych dla pętli sporządzającej poszczególne ekrany.
Linie 30 i 40	— Pętla rysująco-animacyjna. W trakcie przygotowania ekranów wykorzystuje podprogram w liniach 80+100, a do przełączania ekranów podprogram 50+70.
Linie 50+70	— Regulacja szybkości obrotowej.
Linie 80-100	— Rysowanie graniastosłupów. Po skończeniu rysunków podprogram kończy się w linii 110.
Linia 110	— Ustawienie zmiennych. Przygotowanie pętli znajdującej się w liniach 30 i 40 do przełączania ekranów i ustawienie RAMTOP na jego normalnej wysokości.
Linia 120	— Ustawienie szybkości animacji, skoku w pętli i zablokowanie trybu przyciągania uwagi.

Wojciech Przybył

Nie należy wpisywać liter wydrukowanych przed numerem linii programu. NIE są one częścią linii, lecz kodem kontrolnym „Edytora BASIC-a" (zob. „Bajtek" 1/88 lub „Bajtek-Atari").

```
XE 0 REM REGULACJA PREDKOSCI: "START" -
   ZMNIEJSZA, "SELECT" - ZWIEKSZA
OD 9 REM WYBOR WIELOKATA PODSTAWY
JE 10 GRAPHICS 0:? :? :? "  PODAJ LICZBE
   BOKOW N=";:INPUT N:IF N<3 OR N>INT(N)
   THEN 10
CW 19 REM USTALENIE ZMIENNYCH DLA EKRANOW
DF 20 F=106:T=128:D=193:H=80:A=120:B=30:Q
   =35:P=160:KT=360/N:N=KT/16:DEG
TO 29 REM PETLA RYSUJACO-ANIMACYJNA
JQ 30 FOR K=0 TO 8 STEP 8:E=4-K:POKE F,T-
   4*K:FOR J=0 TO 3:IF J THEN D=D+E
GS 40 POKE 54017,D:GOSUB H:NEXT J:NEXT
   K:POKE L,P:GOTO A
QS 49 REM REGULACJA PREDKOSCI
TQ 50 FOR I=1 TO N:IF PEEK(53279)=5 THEN
   IF N>0 THEN N=N-0.3
ZR 60 IF PEEK(53279)=6 THEN N=N+0.3
UT 70 NEXT I:RETURN
KK 79 REM RYSOWANIE GRANIASTOSLUPA
NT 80 GRAPHICS 8+16:COLOR 1
MV 90 FOR I=0 TO 360 STEP KT:X=P+A*COS
   (I+N):Y=Q+B*SIN(I+N):IF I THEN PLOT
   R,S:DRAWTO X,Y:DRAWTO X,Y+A:DRAWTO
   R,S+A
QZ 100 R=X:S=Y:NEXT I:FOR I=1 TO 300:NEXT
   I:N=N+KT/8:IF N<KT THEN RETURN
OV 109 REM USTALENIE ZMIENNYCH DLA
   ANIMACJI
SX 110 L=F:F=561:T=96:H=50:RETURN
NM 120 L=77:P=0:A=30:N=10:GOTO A
```

ATARI 520 ST



Zaprezentowane po raz pierwszy wiosną 1985 r. Atari 520 ST zapoczątkowało całą rodzinę nowych komputerów serii ST. Model ten był również zapowiedzią stopniowego odchodzenia od produkcji komputerów 8-bitowych Atari 800XL/ 130XE, które mimo niskiej ceny przestają być sprzętem poszukiwanym przez użytkowników.

W Atari 520 ST zastosowano mikroprocesor Motorola 68 000 o strukturze 32-bitowej i 16-bitowej szynie danych. Pozwala on na bezpośrednie adresowanie do 16 MB pamięci, lecz praktyczne rozwiązanie innych układów ogranicza jej maksymalny zakres do 4 MB. 520 ST dysponuje pamięcią 512 KB, a jego następcy 1040 ST i MEGA ST mają już odpowiednio 1040 KB i 2 MB. W pamięci ROM mającej 192 KB umieszczono m.in. system operacyjny TOS. GEM (Graphics Environment Manager) łączy technikę okienek, ikon i przewijanych menu wspomagany myszą gwarantują sprawną i wygodną dla użytkownika obsługę komputera. Konstruktorzy Atari 520 ST przewidzieli dołączanie różnych dodatkowych urządzeń: dysku twardego, którego zainstalowanie jest możliwe dzięki układowi DMA (Direct Memory Acces) o szybkości transmisji danych 1,33 Mb/s, drukarki lub instrumentów muzycznych przez interfejs MIDI. Podstawowym rodzajem pamięci masowej jest obecnie dwustronna 3,5 calowa stacja dysków elastycznych SF 314 o pojemności 720 KB.

Użytkownik ma możliwość wyboru monitora monochromatycznego o wysokiej rozdzielczości - 640x400 punktów lub kolorowego 640x300 punktów w 4 barwach i 320x200 w 16 barwach. Ogólnie można korzystać z palety aż 512 kolorów. 3 niezależne, regulowane generatory dźwięku o częstotliwościach przekraczających zakres słyszalności ludzkiego ucha pozwalają naśladować brzmienie instrumentów muzycznych i uzyskiwanie wielu ciekawych efektów akustycznych. Grafika i dźwięk stały się jednym z podstawowych atutów Atari 520 ST i umożliwiły opracowanie setek interesujących programów użytkowych oraz pasjonujących gier.

Liczne zalety Atari 520 ST oraz jego następców sprawiły, że pretenduje on do roli najpopularniejszego komputera domowego. Dostępna dla użytkownika pamięć, urządzenia zewnętrzne i szybki procesor pozwalają konstruować oprogramowanie o walorach przewyższających niejednokrotnie komputery typu IBM PC. Firmy softwarowe dostrzegły możliwości drzemiące w ST i ich ofercie możemy znaleźć dziś wszystkie podstawowe np. języki programowania np. assembler, Basic, Pascal, Modula 2, bazy danych, programy kalkulacyjne, graficzne, muzyczne, a nawet pakiety CAD. Wydawanych jest wiele pism tylko o ST (np. „ATARI ST USER"), a inne (wśród nich „BYTE") przeznaczają dla ST obszerne stałe rubryki. Wzrasta również liczba publikacji książkowych. Kilkanaście tytułów wydawanych rocznie tylko przez wydawnictwo Data Becker jest znamiennym symptomem zainteresowania komputerami Atari ST.

(j.j)

DANE TECHNICZNE

Pamięć

512 KB RAM
192 KB ROM

Struktura wewnętrzna

mikroprocesor 16/32 bitowy Motorola 68000, 8 MHz
osiem 32 bitowych rejestrów danych
dziewięć 32 bitowych rejestrów adresowych
16 bitowa szyna danych
24 bitowa szyna adresowa siedem poziomów przerwań
56 rozkazów, 14 trybów adresowania. 5 typów danych

Przechowywanie danych

bezpośredni dostęp do pamięci 1,33 MB/s
wbudowane gniazdo dla cartridge'a
integralny napęd dysków elastycznych

Złącza

złącze do przyłączenia twardego dysku złącze równoległe drukarki RS 232 (V 24)
złącze DMA dla drukarki laserowej i dysku twardego
gniazdo dla dwóch napędów dysków elastycznych
2 gniazda joysticków
wyjście video dla monitora RGB, telewizora lub monitora monochromatycznego wyjście HF dla telewizora
interfejs MIDI sprzęgający z urządzeniami syntetyzującymi dźwięk

Klawiatura

klawiatura typu qwerty
blok 18 klawiszy numerycznych
klawisze sterujące ruchem kursora
oddzielny procesor obsługujący klawiaturę
dostępne wersje klawiatury angielskiej i niemieckiej

Dźwięk

3 generatory dźwięku
częstotliwość generowanego dźwięku od 30 Hz do 16 KHz 3 kanały
częstotliwość i głośność dźwięku każdego kanału regulowana
dynamiczna kontrola obwiedni dźwięku

Grafika

pamięć obrazu 32 KB
3 tryby rozdzielczości: 320 x 200 punktów w 16 kolorach (najniższa rozdzielczość)
640 x 200 punktów w 4 kolorach (średnia rozdzielczość)
640 x 400 punktów monochromatycznych (najwyższa rozdzielczość) 512 kolorów

System operacyjny TOS z GEM

możliwość jednoczesnego otwarcia do 4 okien
menu rozwijane
biblioteka procedur użytkowych GEM
zegar czasu rzeczywistego

ZOSTAŃ NIEŚMIERTELNYM (3)

Oto następna seria poprawek umożliwiających uzyskanie „wiecznego życia”.

Przypominamy, że do programów w wersji kasetowej można je wprowadzać przy pomocy programu „Zamiana napisów” ("Bajtek 7/87) lub „Nie tylko dla graczy” ("Bajtek” 3/88), a w wersji dyskowej przy pomocy dowolnego monitora dyskowego ("Sherlock”, „Watson”, „Wizard” itd.).

PANTHER

Liczb „żyć” w komórce $A0 — zamienić DEC SAO na LDA $A0 ($C6, SAO na $A5, $A0 — „F” w negatywie, spacja w negatywie na „%” w negatywie, spacja w negatywie).

MOUSE TRAP

Liczba „żyć” w komórce $50 — zamienić DEC $50 na LDA $50 ($C6, $50 na $A5, $50 — „F” w negatywie, „2” na „%” w negatywie, „2”).

MONTEZUMAS REVENGE

Liczba „żyć” w komórce $8D — zamienić DEC $8D na LDA $8D ($C6, $8D na $A5, $8D — „F” w negatywie, „CTRL-M” w negatywie na „%” w negatywie, „CTRL-M” w negatywie).

DONKEY KONG JR.

Liczba „żyć” w komórce $89 — zamienić DEC $89,X na LDA $89,X ($D6, $89 na $B5, $89 — „V” w negatywie, „CTRL-I” w negatywie na „5” w negatywie, „CTRL-I” w negatywie).

PINHEAD

Liczba „żyć” w komórce $B2 — zamienić DEC $B2 na LDA $B2 ($C6, $B2, na $A5, $B2 — „F” w negatywie, „2” w negatywie na „%” w negatywie, „2” w negatywie).

BRUCE LEE

Liczba „żyć” w komórce $26 — zamienić DEC $26 na LDA $26 ($C6, $26 na $A5, $26 — „F” w negatywie, „&” na „%” w negatywie, „&”).

ZAXXON

Liczba „żyć” w komórce $9A — zamienić DEC $9A, X na LDA $9A,X ($D6, $9A na $B5, $9A — „V” w negatywie, „CTRL-Z” w negatywie na „5” w negatywie, „CTRL-Z” w negatywie).

Rozkaz ten występuje w programie dwukrotnie, trzeba więc dokonać zamiany dwa razy.

MOLECULE MAN

Liczba „bomb” w komórce $0605 — DEC $0605 ($CE, $05, $06 — „N” w negatywie, „CTRL-E”, „CTRL-F”).

Liczba „tabletek” w komórce $0606 — DEC $0606 ($CE, $06, $06 — „N” w negatywie, „CTRL-F”, „CTRL- F”).

Liczba „funtów” w komórce $0607 — DEC $0607 ($CE, $07, $06 — „N” w negatywie, „CTRL-G”, „CTRL- F”).

We wszystkich przypadkach należy zamienić rozkaz DEC na LDA ($CE na $AD — „N” w negatywie na „-” w negatywie), przy czym „funty” znajdują się w dwóch miejscach programu.

Tomasz Wiśniewski
Wojciech Zientara

GŁOS O „EDYTORZE BASICA”

W numerze 1/88 „Bajtka” oraz w dodatku „Tylko o Atari” został wydrukowany program „Edytor BASIC-a.

Sposób „odzyskania” programu „czystego”, wpisywanego za pomocą „EDYTORA” jest bardzo niewygodny. (Zapis na nośniku instrukcją LIST; kasowanie pamięci, odczyt instrukcją EN- TER i dopiero właściwy zapis na nośniku).

Programowane przeze mnie zmiany i dodatkowy blok kasowania ułatwiają korzystanie z „Edytora”:

1 — Kod zobrazowany jest w IIWERSE VIDEO, dzięki czemu wyróżnia się od numeru linii (a jest wyświetlany bezpośrednio przed nim).

2 — Po wypisaniu całego programu za pomocą „EDYTORA” wystarczy podać komendę KAS (skrót od — kasowanie) i wszystkie linie „EDYTORA” są wymazywane z pamięci, po czym właściwy program jest powolnie listowany — dla ostatecznego przejrzenia jego treści. Można go potem od razu zapisać na nośniku, bez dodatkowych (niepotrzebnych przecież!) operacji. Niżej podaję kolejno czynności przy przeróbce treści programu.

1.— Wprowadzić do pamięci „Edytor BASIC-a” i uruchomić go.

2. — Zmienić linie 32190 i 32200 (przez *) wpisując zamiast +65 człon +193 (kod — w IN- VERSE VIDEO):

```
32190 KODM=KOD-(KODS*26)+193
32200 KODS=KODS+193
```

3.— Połączyć dwie linie w jedną poprzez dopisanie na początku linii 32090 treści linii 32080:

```
32090 POSITION 2,10:? "CONT":B=VAL(LINIA$)
:POSITION  1,3:? "   ";
```

4.— Dopisać nową linię zamiast starej (przekazanie pracy programu — po komendzie KAS do bloku kasowania „Edytora”)

```
32080 IF LINIA*="KAS" OR LINIA*="kas" THEN
32240
```

5.— Wprowadzić nowe linie stanowiące blok kasowania „Edytora”:

```
32230 REM * KASOWANIE EDYTORA *
32240 ? CHR*(125):POSITION 2,4:FOR 1=0
TO 15:? 32000+1*10:NEXT I:? "CONT"
32250 POSITION 1,2:? " ";:POKE 842,13:
STOP
32260 POKE 842,13:? CHR*(125):POSITION
2,4
32270 FOR 1=16 TO 30:7 32000+1*10:NEXT
I:? "POKE 842,12:? CHR*(125):POKE 622
,255:L.:POKE 622,0":GOTO 32250
32280 REM W celu usuniecia EDYTORA
z pamieci wpisz zamiast linii programu
zlecenie KAS (skrot od KASOWANIE) 32290
REM (c) "Bajtek" 1/88
32300 REM Kasowanie - L. Zgutka
```

Leszek Zgutka

OD REDAKCJI: Pierwsza z opisanych poprawek jest bardzo pomocna w pracy z „Edytorem Basica”. Wiele wątpliwości wywołuje jednak druga propozycja. Atari Basic zapisuje wszystkie zmienne oraz różne „śmieci” powstałe na skutek błędów w tablicy nazw zmiennych. Instrukcje SAVE i CSAVE zapisują tablicę nazw zmiennych wraz z całą zawartością. Natomiast instrukcja LIST oczyszcza program ze zbędnych elementów. Zasadą pracy każdego prograrnisty powinno być zapisywanie gotowego programu najpierw przez LIST.

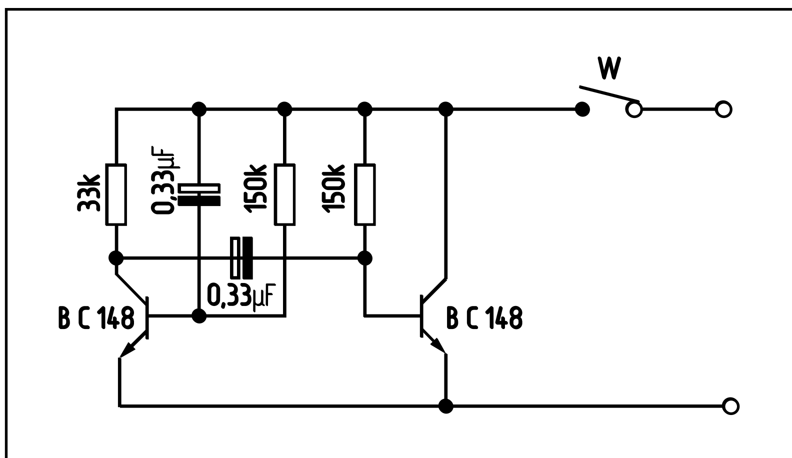
(red.)

AUTO-FIRE

Na giełdach komputerowych i w sklepach Pewexu można zakupić joystick VG-125. Jego droższa wersja VG-125AQ posiada możliwość przełączania na ciągłe strzelanie, tzw. auto-fire.

Zwiększona funkcjonalność nie uzasadnia jednak tak dużej różnicy cen między odmianami tego joysticka. Na szczęście wersja uproszczona posiada płytkę drukowaną, w którą można wlutować elementy według zamieszczonego rysunku. Wyłącznik można zastosować dowolny, np. przełącznik zakresów stosowany w radioodbiornikach bateryjnych. Pewien kłopot mogą sprawić kondensatory. Podczas prób najlepsze okazały się kondensatory tantalowe o pojemności 0,33 µF.

Jerzy Krzyżowski



NOTATNIK NA SCALAKACH

	C-64	C-64	C-64	C-64	C-64	C-64	C-64	C-128	C-128	C-128
NAZWA PROGRAMU	IRC	GEO	FFI	MNG	DMA	SUP	SDA	DMA	SUP	FFI
NOSNIK ORYGINALNY	D/T	DSK	DSK	DSK	D/T	DSK	DSK	DSK	DSK	DSK
DYSKIETKA ZABEZPIECZONA ?	TAK	TAK	TAK	TAK	NIE	TAK	TAK	TAK	TAK	TAK
OPCJA "HELP" DOSTEPNA ?	TAK	NIE	NIE	TAK	NIE	TAK	NIE	TAK	TAK	NIE
BAZA DEMO DOSTEPNA ?	TAK	TAK	TAK	TAK	NIE	NIE	NIE	TAK	NIE	TAK
ZAPIS DANYCH NA KASECIE	TAK	NIE	NIE	NIE	TAK	NIE	NIE	NIE	NIE	NIE
MAKS. ILOSC REKORDOW	1)	?	5000	2000	5205	4)	?	?	4)	5000
MAKS. DLUGOSC REKORDOW	255	?	255	1500	240	1100	760	4096	1100	255
MAKS. ILOSC POL W REKORDZIE	15	?	20	250	8	127	19	100	127	20
MAKS. ILOSC ZNAKOW W POLU	32	?	255	40	30	255	40	255	255	255
UKLAD BAZY PROJEKTOWANY ?	NIE	TAK	NIE	TAK	NIE	TAK	TAK	TAK	TAK	TAK
MAKSYMALNA ILOSC EKRANOW	1	10	1	20	1	1	1	64	4	1
TYP ZBIORU DANYCH	PRG	USR	SEQ	SEQ	REL	SEQ	5)	REL	SEQ	SEQ
POLA ZNAKOWE DOSTEPNE ?	TAK	TAK	TAK	TAK	TAK	TAK	TAK	TAK	TAK	TAK
POLA NUMERYCZNE DOSTEPNE ?	NIE	TAK	TAK	TAK	TAK	TAK	NIE	TAK	TAK	TAK
POLA DLA DATY DOSTEPNE	NIE	NIE	NIE	NIE	TAK	TAK	NIE	TAK	TAK	NIE
INNE DOSTEPNE POLA	NIE	TAK	NIE	NIE	TAK	TAK	TAK	TAK	TAK	NIE
ARYTMETYKA DOSTEPNA ?	NIE	NIE	TAK	TAK	TAK	TAK	6)	TAK	TAK	TAK
PROGRAMOWANIE WARUNKOWE	NIE	NIE	NIE	TAK	NIE	TAK	NIE	TAK	TAK	NIE
RAPORT W ZADANYM UKLADZIE	TAK	TAK	TAK	TAK	NIE	TAK	TAK	TAK	TAK	TAK
WSPOLPRACA Z EDYTOREM TEKSTU	2)	TAK	TAK	TAK	NIE	TAK	TAK	TAK	TAK	TAK
WSPOLPRACA Z ARKUSZEM KAL.	NIE	TAK	NIE	NIE	NIE	NIE	NIE	TAK	NIE	NIE
SORTOWANIE ZBIORU	3)	TAK	TAK	TAK	TAK	TAK	TAK	TAK	TAK	TAK
ROZSZERZANIE ZBIORU	TAK	TAK	TAK	TAK	TAK	TAK	TAK	TAK	TAK	TAK
OPERACJE DYSKOWE	NIE	TAK	TAK	NIE	NIE	TAK	TAK	TAK	TAK	TAK
ZMIANA UKLADU ZBIORU	NIE	TAK	TAK	TAK	NIE	TAK	TAK	TAK	TAK	TAK
WYSZUKIWANIE	TAK	TAK	TAK	TAK	TAK	TAK	TAK	TAK	TAK	TAK
WYDRUK POJEDYNCZEGO REKORDU	NIE	TAK	TAK	TAK	NIE	TAK	TAK	TAK	TAK	TAK
KORZYSTA Z DYSKU SYSTEMOWEGO	NIE	TAK	NIE	TAK	NIE	TAK	NIE	TAK	TAK	NIE
PRACA NA DWIE STACJE MOZLIWA	NIE	TAK	TAK	NIE	?	TAK	NIE	NIE	TAK	TAK
FORMATOWANIE DYSKIETKI	NIE	NIE	NIE	TAK	TAK	TAK	TAK	TAK	TAK	TAK
TRYB PRACY (40/80 ZNAKOW)	40	40	40	40	40	40	40	80	80	80

UWAGI:

1) 121 przy 255 znakach w rekordzie.
2) Tworzy ze zbioru plik typu SEQ.
3) Pozwala na wprowadzanie rekordu pomiedzy inne.
4) Ilosc rekordow ograniczona jedynie pojemnoscia stosowanej stacji.
5) Dane sa wpisywane bezposrednio w sektory dyskietki.
6) Operacje arytnetyczne nozliwe poza polami.

Programy GEOFILE i STARDATEI sa elektronicznymi kartotekami.
SUPERBASE umozliwia ukladanie programow we wlasnym jezyku.

Skroty nazw programow:

IRC	INSTANT RECALL	MASTERTRONICS
GEO	GEOFILE 64	BERKELEY SOFTWORKS
FFI	FLEET FILER (64 i 128)	PSI
MNG	The MANAGER 64	CBM
DMA	DATA MANAGER (64 i 128)	TIMEWORKS, INC.
SUP	SUPERBASE (64 i 128)	PRECISION SOFTWARE, INC
SDA	STAR DATEI	SYBEX

Gdybyśmy zapytali kompletnego laika komputerowego, z czym kojarzy mu się ta maszyna, to jestem pewny, że wymieniłby bazy danych zaraz po zastosowaniach matematycznych. W rzeczywistości wielu z nas przysparza sobie tylko sporo dodatkowej pracy usiłując zelektronizować swój notesik czy notatnik z adresami.

Wielu posiadaczy mikrokomputerów zupełnie niepotrzebnie marnuje swój czas zakładaniem baz danych zawierających 25 czy 40 rekordów. Niepotrzebnie dlatego, ponieważ w rzeczywistości znacznie szybciej odszukamy adres kumpla czy jego telefon właśnie w kaldendarzyku. Być może twierdzenie to wyda się komuś nieprawdopodobne, ale jest to prawda. Policzmy czas, jaki zajmie nam wyjęcie kalendarzyka, wybranie odpowiedniej litery w skorowidzu i wyszukanie żądanej informacji; czas ten nie zajmie nam więcej niż minutę, góra dwie. Ta sama operacja przy wykorzystaniu komputera będzie minimum dwa razy dłuższa — trzeba komputer włączyć, wczytać program, zmienić dyskietki i wykonać jeszcze kilka operacji dodatkowych. Z tego wniosek, że komputer jest wolniejszy od człowieka i... w tym wypadku jest to prawda.

Zakładanie jakiejkolwiek bazy danych ma sens jedynie wtedy, gdy nasz zbiór informacji jest bardzo duży i chcemy w uporządkowany sposób odczytywać zapisane w nim informacje. Przykładem może być tu np. zbiór obejmujący dane o dużym księgozbiorze, dane tematyczne o artykułach w iluś tam rocznikach naszego ulubionego czasopisma, dane o znaczkach itp. Czasami warto w tym celu napisać własny program lub skorzystać z już gotowych.

Poniższa tabela zawiera porównanie kilku programów dostępnych obecnie na rynku (czytaj — giełdzie) do komputerów C-64 i C-128. Oczywiście nie byłem w stanie podać wszystkich, gdyż potrzebowałbym na co najmniej pół numeru BAJTKA; starałem się więc wybrać te programy, które oferują dość bogate możliwości. W wypadku C-128 nie wspominałem o najbardziej uniwersalnym — dBASE II firmy AshtonTate, gdyż po pierwsze działa on pod kontrolą systemu CP/M, a po drugie jest to program dość dobrze znany. Do swoich prywatnych zbiorów (raptem 4) używam tylko i wyłącznie dBASE II, gdyż system ten daje rzeczywiście ogromne możliwości, które kiedyś niewątpliwie w BAJTKU opiszę.

Z programów proponowanych dla C-64 warto wspomnieć o SUPERBASE 64 oraz THE MANAGER, dających użytkownikowi posmak tego, czym powinna się dobra baza danych charakteryzować. DATA MANAGER dla C-128 jest częścią pakietu zintegrowanego składającego się z edytora tekstu (niestety bez polskich liter) oraz arkusza kalkulacyjnego, co znacznie powiększa moim zdaniem wartość tego programu. SUPERBASE 128 jest także hitem dla tego modelu komputera i współpracuje z programem SU- PERSCRIPT (edytor tekstu) również dostępnym w wersji 128. Brak programów przeznaczonych dla C-16 i PLUS/4 Czytelnik zawdzięcza głównie brakowi informacji o odnośnym oprogramowaniu do tych modeli (pomijając oczywiście bazę zapisaną w ROM Commodore PLUS/4). A może Czytelnicy — właściciele pomogą mi zdobyć informacje o bazach danych dla rodziny C- 16, C-116 i PLUS/4?

Klaudiusz Dybowski

SEQ, REL — I FERAJNA —

Skróty te są zapewne dobrze znane posiadaczom stacji dysków 1541 czy 1570; nie każdy jednak wie dokładnie o co chodzi. Ponieważ tematem przewodnim tego numeru „Bajtka" są bazy danych, warto więc przyjrzeć się tym skrótom nieco bliżej.

Tytułowa „ferajna" ma swoje źródło w tym, że tak naprawdę istnieją jeszcze trzy dodatkowe typy zbiorów (plików) jakie można utworzyć, jeżeli dysponuje się stacją dysków — DEL, PRG i USR. Pierwszym z nich nie będziemy się zajmować w ogóle, gdyż oznacza on plik skasowany. Cztery pozostałe typy zbiorów — PRG, SEQ, REL i USR —- to typy zbiorów, z którymi masz do czynienia w zasadzie na co dzień.

Najpopularniejszym z nich jest zbiór programowy oznaczany w katalogu dyskietki jako PRG. Od pozostałych różni się on tym, że pozwala zapisywać rozkazy i instrukcje BASIC w postaci jednego bajtu. Taki zapis pozwala nam niewątpliwie zaoszczędzić sporo miejsca na dyskietce. W bazach danych jednak, zbiory typu programowego (PRG) są wykorzystywane dość rzadko, gdyż bardzo rzadko istnieje potrzeba zapisywania w nich cyfrowych postaci instrukcji i poleceń BASIC.

USR to skrót oznaczający zbiór przeznaczony dla użytkownika (USeR). Jego struktura jest w zasadzie identyczna ze strukturą zbioru sekwencyjnego (SEQ) z tym, że znaczek „&" wpisany w tytule pozwala nam (jeżeli ktoś umie programować w języku maszynowym) na automatyczne wczytanie zawartości takiego zbioru do pamięci RAM stacji dysków i (również automatyczne) uruchomienie i wykonanie programu zapisanego w takim zbiorze. Podobnie jak zbiór PRG jest on dość rzadko wykorzystywany do przechowywania danych.

Przeczytałeś w swoim życiu już niejedną książkę. Przykład książki ilustruje doskonale zbiór sekwencyjny oznaczany w katalogu dyskietki (directory) jako SEQ. Podczas tworzenia takiego zbioru dane są zapisywane jedna za drugą w dokładnie takiej samej kolejności, w jakiej wpisujesz je do komputera. Zapis tego rodzaju ma zarówno wady jak i zalety. Zaletą jest tu niewątpliwie fakt, że umieszczając dane w zbiorze jedna za drugą nie „marnujesz" nawet jednego bajtu z dyskietki — gdy wpisałeś jedno nazwisko, zaraz za nim jest zapisywane następne. Stąd też zbiory takie są często używane tam, gdzie nie istnieje konieczność szybkiego wyszukiwania danych. Ten typ zbioru jest też jedynym możliwym do zastosowania w wypadku pamięci taśmowych. Wadą zbiorów sekwencyjnych jest z kolei powolność ich przeszukiwania — aby odczytać setne nazwisko zapisane czy to na dyskietce czy to na taśmie, komputer musi najpierw odczytać 99 nazwisk zapisanych wcześniej. Dlatego też do zapisu zbiorów danych o dużej ilości rekordów (jeden rekord oznacza w tym przypadku jedno zapisane nazwisko) pliki typu SEQ nie nadają się zupełnie — olbrzymia większość czasu to czas zmarnowany na odczytywanie niepotrzebnych nam informacji.

Wady tej pozbawione są z kolei zbiory określane jako REL (RELative) czyli zbiory o dostępie bezpośrednim. Zbiory takie możemy tworzyć jedynie na dyskietce, ponieważ tylko w stacjach dysków głowica odczytująca jest ruchoma i przesuwa się nad nośnikiem magnetycznym. Taki sposób odczytu umożliwia nam „przeskakiwanie" w dowolne miejsce dyskietki i odczytywanie zapisanych w niej informacji. Nieruchoma głowica w magnetofonie kasetowym nie jest w stanie zapewnić nam tego komfortu.

Niejednokrotnie odbierałeś już pewnie z zakładu fotograficznego wywołany film ze slajdami. Aby przejrzeć po kolei czy zdjęcia Ci się udały, rozwijałeś rolkę i klatka po klatce przeglądałeś każde prześrocze; jak się domyślasz taki sposób odczytu informacji jest właśnie przykładem odczytu sekwencyjnego.

Zupełnie inaczej wygląda sprawa, gdy wszystkie slajdy są już pocięte, włożone w ramki i znajdują się w pudełku czy odpowiednim magazynku. Gdy chcesz obejrzeć jeden z nich nie musisz już przeglądać wszystkich — wystarczy jeżeli każdy z nich ponumerujesz i wyciągniesz z pudełka prześrocze o określonym numerze. Zamiast więc przeszukiwać całe pudełko czy magazynek, wyjmujesz tylko jeden slajd bez konieczności oglądania pozostałych (co miałoby miejsce, gdybyś rozwinął nie- pocięty pozytyw i szukał tego jednego slajdu klatka po klatce). W bardzo podobny sposób odczytywane są właśnie zbiory o bezpośrednim dostępie. Dzięki odpowiedniej tabeli tworzonej na dyskietce w chwili ich zapisu DOS (dyskowy system operacyjny) jest w stanie obliczyć, gdzie znajduje się rekord o określonym numerze i może go odczytać nie marnując czasu na odczyt innych.

Jak z tego wynika, stosowanie takich zbiorów we własnych bazach danych gwarantuje nam dużą szybkość wyszukiwania informacji. Wadą jest tu niestety pewne „marnotrawstwo" — każdy rekord musi składać się z określonej z góry liczby znaków (maksymalnie 254). innymi słowy, jeżeli na nazwisko w każdym rekordzie przeznaczyłeś powiedzmy 10 bajtów, a składało się ono z trzech liter, to 7 bajtów przepadło już nam bezpowrotnie — zostaną one zapisane jako puste.

Artykuł ten nie ma celu zapoznania Cię Czytelniku z zasadami programowania i wykorzystywania poszczególnych typów zbiorów. Na dyskietce demonstracyjnej dołączonej do każdej stacji przy jej zakupie znajdują się dwa programy — „SEQ FILE DEMO" oraz „REL FILE DEMO", które w przystępny i zrozumiały sposób wprowadzą Cię w arkana stosowania tych zbiorów. Zamiast tego chciałbym tu raczej napomknąć kilka słów na temat pewnych sztuczek i usprawnień jakie można i warto czasami zastosować projektując swoją własną bazę danych.

Najszybszym sposobem selekcji i wyszukiwania danych jest umieszczenie ich w pamięci operacyjnej komputera — np. w tablicy. Taka metoda może być zastosowana jednakże jedynie w wypadku zbiorów niezbyt wielkich, powiedzmy ok. 1000—2000 pozycji. Jeżeli przewidujesz, że Twój zbiór danych będzie większy, to warto być może pokusić się o jego podział; przykładowo, gdy tworzysz zbiór danych obejmujący listę posiadanych przez Ciebie programów, wyszukiwać je będziesz najczęściej po tytule. Ponieważ zaś każdy tytuł zaczyna się na jakąś literę (lub cyfrę) opłaca się czasami utworzyć szereg oddzielnych zbiorów, w których zapisywać będziesz tytuły zaczynające się na daną literę; otwieranie i zamykanie dostępów do tych plików można zrealizować za pomocą trzech linii w BASIC:

```
10 INPUT „Wpisz tytuł programu": A$:B$ = LEFT$
(A$,1)
20 C$ = B$+".BAZA,S.R": REM nazwa zbioru np. A.
BAZA, B.BAZA,S,R: B.BAZA itp.
30 OPEN 5,8,5, (C$)
```

Program-baza powinien być oczywiście jak najkrótszy co zwiększa także pojemność pamięci przeznaczonej na naszą ewentualną tablicę. W rzeczywistości wszystkie taśmowe bazy danych wczytują najczęściej całą zawartość zbioru do przygotowanej wcześniej tablicy; dzięki temu wzrasta szybkość operacji i wyszukiwania, można także bez przeszkód zbiór uzupełniać. Gdy dokonamy już wszystkich niezbędnych korekt i poprawek zawartość tej tablicy zapisujemy na taśmie czy dyskietce za pomocą:

```
1000 OPEN 1,1,2, „BAZA Danych":CR$=CHR$(13)
1010 FOR X = 0TO N : REM N = Ilość rekordów
1020 PRINT # 1.AS(N); CR$;B$(N);CRS;CS(N)
1030 NEXT X
1040 CLOSE 1
```

Gdy zapisujemy na taśmie czy dysku nasze dane warto pokusić się o wpisanie jeszcze jednej informacji — całkowitej ilości nazwisk czy tytułów w naszej bazie zawartych. Jeżeli zapis taki poprzedzi zapis danych, to unikniemy konieczności deklarowania tablicy „na wyrost" podczas odczytu; innymi słowy będziemy w stanie od razu określić, że należy utworzyć tablicę dla np. 297 wartości. W takim wypadku należy dodać do powyższego programu;

```
1002 * PRINT 1,N
```

Gdy zamierzamy odczytywać dane ze zbioru wystarczy teraz skorzystać z następującego programu:

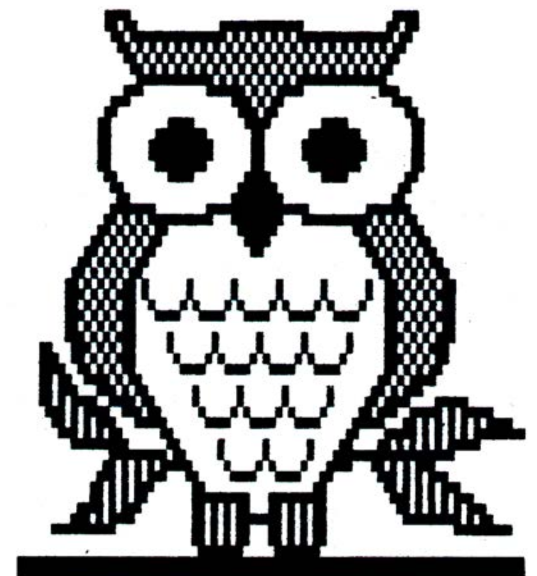
```
2000 OPEN 1,1,0, „BAZA DANYCH"
2010 INPUT 1,N
2020 DIM A$(N), B$(N), C$(N)
2030 FORI=0TON
2040 IF ST = 64 GOtO 2999
2050 INPUT # 1,A$(I).B$(I),C$(I)
2060 NEXT i
2999 CLOSE 1
```

Linia 2040 pozwoli nam określić precyzyjnie koniec zbioru i zamknąć dostęp do niego w chwili odczytania ostatnich danych.

Odrębnym zagadnieniem jest tu dobranie odpowiednio szybkich algorytmów sortujących i wyszukujących. Znaczne zwiększenie szybkości wyszukiwania danych umożliwia nam albo zastosowanie procedur w języku maszynowym, albo (w nieco mniejszym stopniu) kompilacja całego programu. Ciekawym rozwiązaniem może tu być tzw. system wyszukiwania binarnego. W dużym skrócie polega on na szybkim dzieleniu zbioru przez dwa. Załóżmy, że zapisaliśmy jako 84 pozycję nazwisko MALINIAK i staramy się je wyszukać stosując tę metodę (zakładam też, że nazwiska są ułożone w kolejności alfabetycznej). Najpierw dzielimy zbiór na dwie połówki — od rekordu 1 do 50 i 51 do 100. Sprawdzamy ostatni rekord pierwszej połowy — zawiera on nazwisko np. Kowalski. Komputer wie już, że trafiliśmy za blisko. Zbiór jest więc dzielony znowu na dwie połowy — tym razem od rekordu 51 do 75 i 76 do 100. Po przetestowaniu rekordu 75 wiemy na pewno, że poszukiwany przez nas rekord znajduje się gdzieś w drugiej połówce czyli pomiędzy rekordami 76—100. Wykonywany jest więc następny podział i w rezultacie zamiast przeszukiwać wszystkie 100 rekordów po kolei po trzecim podziale ograniczamy ilość rekordów do przeszukania do zaledwie 12 — od numeru 76 do 88. Nic więc dziwnego, że algorytm ten jest bardzo często stosowany w firmowych bazach danych oferowanych dla komputerów Commodore.

Ostatnio pojawiły się w prasie zagranicznej dość niepokojące artykuły dotyczące przede wszystkim zbiorów typu REL i stacji dysków 1571. Zgodnie z testami przeprowadzonymi przez renomowane czasopisma okazało się, że niektóre procedury DOS nie są dopracowane do końca i w efekcie mogą wystąpić pewne problemy z zapisem rekordów o określonych numerach czy też znaczne spowolnienie zapisu danych na drugiej stronie dyskietki. Problemy te zostaną opisane w oddzielnym artykule. Ja natomiast polecałbym posiadaczom tych stacji korzystanie z firmowych programów, z których większość (np. DATA MANAGER 128) omija te rafy dzięki językowi maszynowemu. Innym rozwiązaniem tego problemu może też być skorzystanie z w pełni profesjonalnego programu dBASE II pracującego pod kontrolą systemu CP/M.

Klaudiusz Dybowski



KOMBINACJE

Oddajemy Wam do użytku program obliczający kombinacje z powtórzeniami, ze wzoru [(n!/(k!*(n-k)!)]. Program ten został napisany na Commodore 64, ale działa na większości komputerów, bez konieczności przeróbek.

Posiadacze Commodore'ów nie muszą wpisywać poleceń LET, ale tylko następujące po nich założenia oraz nie muszą w linii 10 wpisywać DIM A$(3).

Za N i K można podstawiać nawet bardzo duże liczby, ale różnica pomiędzy nimi nie może być zbyt wielka, gdyż wynik działania wykroczy poza obszar zakresu liczbowego komputera (dla różnych komputerów zakres ten jest odmienny). Jeżeli wynik obliczeń kombinacji N po K będzie przekraczał maksymalną dopuszczalną wartość obszaru liczbowego Waszego komputera, program zostanie przerwany i ukaże się komunikat błędu (wykroczenia poza zakres liczbowy).

Mamy nadzieję, że program ten oszczędzi Wam żmudnej i czasochłonnej pracy.

(K&B)

```
10 DIM A$(3):PRINT:PRINT"
K&B":PRINT
20 PRINT"          ROBERT PASZKOWSKI
&":PRINT
30 PRINT"        KRZYSZTOF
CHASZCZEWSKI"
40 PRINT:PRINT"
PREZENTUJA:":PRINT:PRINT
50 PRINT"     PROGRAM TEN OBLICZA WED£UG
WZORU             N!/(K!*(N-K)!)"
60 PRINT:PRINT" CZYLI ILO$C KOMBINACJI N
PO K (N>K)"
70 PRINT:PRINT" OZNACZENIE N! OZNACZA N
SILNIA":PRINT:PRINT"WPROWADZ N"
80 INPUT A:PRINT" WPROWADZ K":INPUT C:IF
C>A THEN RUN
90 LET  X=C:LET L=A-C:DIM A(L)
100 FOR K=1 TO L:LET X=X+1:LET
A(K)=X:NEXT K
110 LET S=1:FOR K=1 TO L:LET
S=S*A(K):NEXT K:PRINT:LET T=0: LET Y=A-
C:DIM M(Y)
120 FOR U=1 TO Y:LET T=T+1:LET
M(U)=T:NEXT U:LET I=1
130 FOR U=1 TO Y:LET I=I*M(U):NEXT
U:PRINT: PRINT"     A OTO WYNIK -> "S/I
140 PRINT:PRINT:PRINT"    CZY JESZCZE
RAZ ? (T/N)"
150 INPUT A$:IF A$="T" THEN RUN
160 IF A$="N" THEN STOP
170 IF A$<>""  THEN GOTO  150
```

PRZEDSTAWIAMY WARSAW BASIC (6)

PROCEDURY

W SYSTEMIE WB

Przy pierwszym kontakcie z edytorem ekranowym interpretera cieszy sam fakt, że komputer może nam służyć jako maszyna do pisania na ekranie (w różnych miejscach różne znaki). Na wyższym etapie tworzenie napisów może służyć do wydawania rozkazów do natychmiastowego wykonania (tryb bezpośredni lub ekranowy). Wykonywanie rozkazów, poprzedzone zapisaniem ich do pamięci operacyjnej, wymaga jeszcze większych umiejętności.

Na tym etapie mogą już powstać programy, służące do rozwiązywania nawet bardzo dużych problemów. Zachowanie ich dla potomności lub powtórnego użycia wymaga pamięci zewnętrznej do ich składowania. Takie wykorzystanie pamięci zewnętrznej, to etap czwarty według naszej klasyfikacji.

Wyobraźmy sobie, że rozwiązaliśmy już kilka większych problemów i każdy z programów zapisaliśmy w pamięci masowej. Może się zdarzyć, że nowy problem wymaga wykonania kolejno tych kilku zadań lub ich fragmentów. Można oczywiście połączyć w trybie bezpośrednim ze sobą wcześniej napisane programy i uruchomić całość. Pomijamy tutaj uciążliwości związane z tego rodzaju łączeniem. Może okazać się jednak, że konieczność wyeliminowania powtarzających się nazw zmiennych i uzgodnienia numerów linii skutecznie utrudnia (a nawet uniemożliwia) poprawne wykonanie tego zadania Chciałoby się, aby wszystkie te uciążliwe prace redakcyjne można było pominąć i po prostu w programie wywołać kolejno zadania, do wykonania ź pamięci zewnętrznej — z taśmy lub dyskietki.

Można tego dokonać odpowiednio projektując wykorzystanie pamięci zewnętrznej w trybie programowym. Ten najwyższy w naszej klasyfikacji etap wykorzystania interpretera nie jest możliwy do osiągnięcia przy pomocy większości znanych nam translatorów, zaimplementowanych na mikrokomputerach domowych. Wyjątkiem jest tutaj interpreter języka Warsaw BASIC.

Aby wyeliminować kłopoty z uzgodnieniem numerów linii, nasze zadanie podzielimy na program główny i wywoływane w tym segmencie procedury. Jeżeli więc zadbamy o to, aby procedury były ładowane do pamięci w inne miejsce niż program główny, to problem prenumerowania linii zniknie.

Problem uzgadniania nazw zmiennych zniknie, jeśli program główny i procedury będą miały rozdzielne zbiory zmiennych. Jak już o tym wspomnieliśmy w odcinku „Organizacja pamięci" (Bajtek Nr 4/88), interpreter CBM V2 jest na tyle elastyczny, że taka propozycja jest możliwa do wykonania.

Rozwiązanie według tej recepty naszego zadania wymaga odpowiedniej organizacji pamięci operacyjnej. Trzeba stworzyć mechanizm podziału pamięci na poziomy. Poziom zerowy przyporządkujemy tej części pamięci, którą zajmuje program główny i jego zmienne. Bezpośrednio za zbiorem zmiennych programu głównego jest miejsce na poziom 1, na którym system będzie umieszczał procedury wywoływane przez program główny. Nic nie stoi na przeszkodzie by procedury z poziomu 1 mogły wywoływać inne procedury. Poziom 2 utworzymy bezpośrednio po zbiorze zmiennych poziomu 1. poziom 3 po zmiennych poziomu 2 itd.

Podział taki, poza wymienionymi już udogodnieniami, dodatkowo umożliwia wykonanie długich programów, które w całości nie zmieściłyby się w pamięci operacyjnej. W tym przypadku kolejne zadania i zbiory ich zmiennych nakładają się bowiem w pamięci, tzn. wywołanie następnej procedury powoduje jej wczytanie z pamięci zewnętrznej i umieszczenie w miejscu poprzedniej. Tym samym procedury tworzą w pamięci zewnętrznej bibliotekę, z której można korzystać w każdym programie w sposób automatyczny — bez żadnych zabiegów redakcyjnych. Aby użyć procedury w programie trzeba tylko znać jej nazwę, pod którą jest ona przechowywana w bibliotece oraz znaczenie jej parametrów. Tak więc korzystanie z procedur nie różni się niczym, poza miejscem przechowywania, od korzystania ze słów kluczowych interpretera, które są obsługiwane przez procedury rezydujące na stałe w pamięci operacyjnej.

W znanych nam implementacjach języków proceduralnych, na mikrokomputerach domowych, korzystanie z biblioteki procedur wymaga dołączania ich do programu na etapie edycji. Często procedury muszą być w trakcie takiego dołączania odpowiednio dostosowane, a całość musi być przed wykonaniem skompilowana. Opisany powyżej mechanizm wykorzystania biblioteki procedur w trybie interpretacji wyróżnia zatem, jak nam się wydaje, Warsaw BASIC spośród innych translatorów.

Poza wspomnianymi już sposobami używania procedur, takimi jak nakładkowanie i zagnieżdżanie, Warsaw BASIC umożliwia programowe tworzenie pakietów procedur w pamięci operacyjnej. Wystarczy w tym celu wskazać nazwy procedur, które mają być załadowane z pamięci zewnętrznej i połączone. Ewentualne powtarzanie się numerów linii w wybranych do pakietu procedurach nie przeszkadza w jego tworzeniu Zestawianie procedur umożliwia wielokrotne wywoływanie ich z pakietu w dowolnej kolejności, bez potrzeby ładowania z pamięci zewnętrznej przy każdym kolejnym wywołaniu.

W najbardziej rozbudowanej wersji WB 3.2 pakiet procedur można umieścić w RAM-dysku. Procedury w nim umieszczone mogą wywoływać się wzajemnie i ich nazwy można traktować jako nowe słowa kluczowe interpretera. Można je także używać w trybie bezpośrednim bez potrzeby ładowania z pamięci zewnętrznej i stosować rekursję. Głębokość zagnieżdżenia ograniczona jest tylko pojemnością stosu. Zawartość RAM-dysku, której nie niszczy nawet instrukcja NEW, może być oczywiście zmieniana w trakcie wykonywania programu. Pakiet procedur przy nakładkowaniu zachowuje się tak jak pojedyncza procedura, tzn. na procedury z pakietu można nałożyć procedurę wywoływaną z pamięci zewnętrznej. Procedury w RAM-dysku są zabezpieczone przed takimi nakładkami.

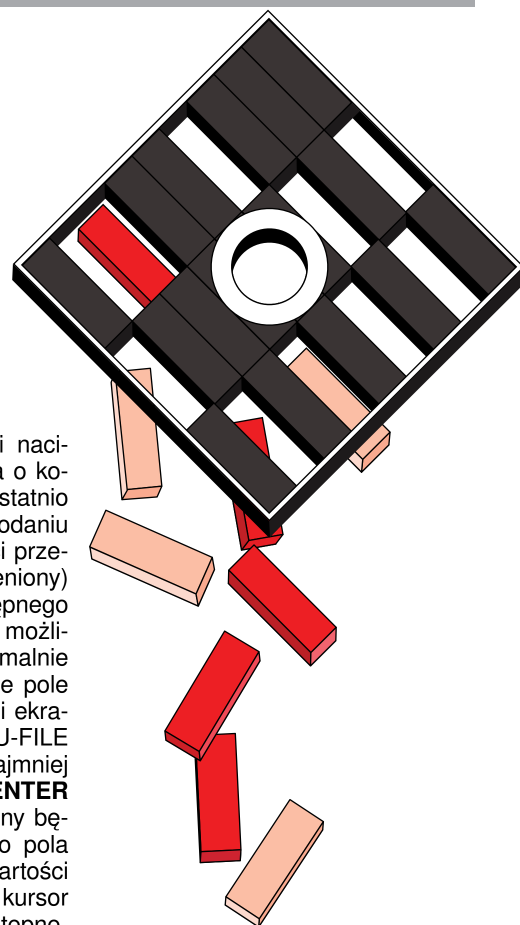
Do wywołania procedury z dyskietki w systemie WB wystarczy podanie jej nazwy i listy parametrów. Translator WB po napotkaniu w programie ciągu znaków, których nie może zinterpretować jako kodu instrukcji, traktuje ten ciąg jako nazwę procedury. Następnie sprawdza czy procedura o tej nazwie znajduje się w pamięci operacyjnej na poziomie o jeden wyższym. Jeśli jej tam nie ma, to sprawdza czy jest założony RAM-dysk. Jeśli jest RAM-dysk, to w nim poszukiwana jest procedura. Jeśli nie ma RAM-dysku lub nie ma w nim poszukiwanej procedury, to translator sprawdza czy jest ona w dostępnej pamięci zewnętrznej. Jeśli procedury nie ma na dyskietce znajdującej się aktualnie w stacji dysków, to proponuje zmianę nośnika. Dopiero po włożeniu właściwej dyskietki, poszukiwana procedura jest wczytywana do pamięci operacyjnej i program jest kontynuowany.

Warsaw BASIC zawiera narzędzia, które wraz z RAM-dyskiem umożliwiają efektywne programowanie języków problemowo zorientowanych z własnymi strukturami danych, tworzonymi w oparciu o tablice ze zmiennymi wymiarami. Nazwy procedur są słowami kluczowymi tych języków. Nasz system relacyjnej bazy danych „eBase WB" jest przykładem takiego języka. Wprowadzone w nim własne typy zmiennych: zmienne rekordowe, formatowe i indeksowe są zaimplementowane tak, że używają tablic tekstowych i numerycznych o zmiennych wymiarach w sposób niewidoczny dla użytkownika. Te nowe zmienne umożliwiają m.in. wykonywanie takich operacji obsługi bazy danych jak przygotowanie raportów, sortowanie i wyszukiwanie informacji według reguł dowolnego alfabetu, a w szczególności według reguł alfabetu polskiego. Nasza biblioteka Warsaw BASICa zawiera również języki problemowo zorientowane na rachunek macierzowy — „Matrix WB" i zespolony — „Complex WB".

W następnych odcinkach przedstawimy dalszą rozbudowę interpretera o podprogramy, które umożliwiają dynamiczny podział pamięci na poziomy, wywołanie procedury z pamięci zewnętrznej z załadowaniem lub bez (jeśli procedura jest juz w pamięci), przekazywanie parametrów z programu wywołującego do procedun/ przez wartość i z procedury do programu głównego przez nazwę procedury oraz nakładkowanie procedur.

Krzysztof Gajewski
Bogusław Radziszewski

JEDYNA SZANSA: VU-FILE



Czy użytkownik ZX Spectrum może założyć bazę danych? Oczywiście, że tak. Pod warunkiem jednak, iż nie zniechęci go zapisywanie i odtwarzanie danych z taśmy magnetofonowej. Do zbioru swych programów dołączyć więc powinien VU-FILE firmy PSION SOFTWARE.

Program VU-FILE służy do zarządzania zbiorami danych definiowanych przez użytkownika. Poprzez pracę w trybie konwersacyjnym (język angielski) jest bardzo łatwy w obsłudze i umożliwia szybkie naprawienie ewentualnych błędów i pomyłek popełnionych przez obsługującego bazę danych. Oczywiście wielkość obszaru pamięci przeznaczonego na dane nie zadowoli profesjinalny (pozostaje około 34 KB) ale i amator powinien zadbać o właściwe wykorzystanie każdego bajtu.

Logiczna organizacja zbiorów obsługiwanych przez VU-FILE jest następująca. Zbiór danych podzielony jest na mniejsze jednostki organizacyjne zwane rekordami, te zaś składają się tzw. pól. Program umożliwia dostęp do pojedynczych rekordów poprzez wyświetlanie ich na ekranie. Jednorazowo można wyświetlić jeden rekord. Obszar przeznaczony do wyświetlania rekordów jest prostokątem o 20 wierszach i 32 kolumnach. W tym obszarze wyświetlać można dowolne rekordy ze zbioru danych, wyszukiwane za pomocą rozkazów programu. Najmniejszą rozpoznawalną jednostką organizacyjną jest pole, które może składać się z sekwencji znaków alfanumerycznych o długości nie przekraczającej 32 znaki. W skład pojedynczego rekordu może wchodzić maksymalnie 30 pól. Wszystkie rekordy w zbiorze danych mają taką samą budowę, tzn. składają się z takiej samej ilości pól. Jednak długości tych samych pól w różnych rekordach mogą być różne. Długość pojedynczego pola nie jest ustalona (jedynym ograniczeniem jest maksymalna długość 32 znaków), a VU-FILE dynamicznie przydziela pamięć dla każdego pola, w zależności od jego długości. Zatem ilość zajętej pamięci przez konkretny rekord nie jest z góry ustalona, ale zależy od rozmiarów poszczególnych pól wchodzących w skład rekordu (jeden znak pola zajmuje 1 bajt pamięci).

Rekordy wyświetlane są w formacie zdefiniowanym przez użytkownika. Oprócz zawartości poszczególnych pól, na ekranie mogą być wyświetlane elementy powtarzające się we wszystkich rekordach a zapamiętywane jednorazowo przez VU-FILE w osobnym zbiorze (prowadzi to do oszczędności pamięci). Mogą to być tytuły, nagłówki, nazwy pól, podkreślenia, tabele itp. Ich postać i układ są również definiowane. Oprócz formatu używanego do wyświetlania rekordów na ekranie, można zdefiniować zupełnie inny format przeznaczony do wydruku ich na drukarce. Z reguły jest on bardziej zwarty w celu oszczędności papieru. Komunikaty i zlecenia w języku angielskim mogą być nie dla wszystkich zrozumiałe, zatem podaję krótki zarys obsługi programu.

Po wczytaniu bazy mamy do wyboru: „E" — rozpoczęcie pracy baz danych lub „L" — wczytanie zbioru danych z taśmy.

Wybranie przycisku „E" jest równoznaczne z definiowaniem od początku formatu rekordów, położenia pól i wpisywaniem rekordów począwszy od pierwszego. Wyświetlony zostaje nowy nagłówek (**RECORD LAYOUT**...) oraz czysty ekran z migającym w lewym górnym roku kursorem. Wolna część ekranu to obszar obrazu rekordu. Jest ona przeznaczona do wyświetlania zawartości pojedynczego rekordu. Przesuwając kursor za pomocą strzałek (na klawiaturze) i wpisując tekst w dowolnych miejscach, ustalić można wszystkie wymagane stałe elementy, które będą pojawiać się przy wyświetlaniu każdego rekordu. W dowolnym momencie, po wciśnięciu klawisza **EDIT** ukaże się okno z wartościami atrybutów. Nazwy atrybutów oraz ich dopuszczalne wartości są zgodne z nazwami i dopuszczalnymi argumentami odpowiednich instrukcji języka BASIC ZX Spectrum, Wartości przedstawione na ekranie można zmieniać używając odpowiednich przycisków numerycznych i **ENTER**. Atrybuty raz ustawione, pozostają aktywne aż do następne ich modyfikacji.

Po wpisaniu wszystkich żądanych tekstów należy wcisnąć STOP, co powoduje automatyczne przejście do funkcji pozwalającej na ustalenie pozycji wyświetlania poszczególnych pól rekordu. Nowy nagłówek (**DATA FIELDS**) informuje o tym, że we wprowadzonym wzorcu rekordu, przy pomocy kursorów i **ENTER** ustalamy początkowe pozycje każdego pola rekordu.

Po przesunięciu kursora i naciśnięciu **ENTER** program pyta o kolor tła (**PAPER**) i liter (**INK**) ostatnio ustalonego pola danych. Po podaniu odpowiednich dwóch wartości przesuwamy kursor (nieco zmieniony) i definiujemy położenie następnego pola rekordu. W ten sposób możliwe jest zdefiniowanie maksymalnie 30 pól, przy czym pojedyncze pole musi mieścić się w jednej linii ekranu. Gdy wciśniemy **STOP**, VU-FILE zażąda wprowadzenia przynajmniej jednego nowego rekordu (**ENTER A RECORD**). Kursor ustawiony będzie na początku pierwszego pola rekordu. Po wpisaniu zawartości pola i wciśnięciu **ENTER**, kursor przechodzi na początek następnego pola. Po wyczerpaniu wszystkich pól, rekord automatycznie dołączany jest do zbioru danych, a kursor znowu ustawia się na pierwszym polu co umożliwia wpisanie kolejnego rekordu. Jeżeli chcemy przerwać wpisywanie rekordów, to w momencie, gdy kursor wskazuje na pierwsze pole, należy wcisnąć STOP. VU-FILE przechodzi wówczas do działania w trybie zleceń, wyświetlając nagłówek z zestawem następujących zleceń (**COMMANDS ARE**...):

ENTER — służy do dołączania nowego rekordu do zbioru danych (patrz opis powyżej).

ALTER — modyfikowanie zawartości aktualnie wyświetlonego rekordu. Po wywołaniu zlecenia kursor ustawia się na początku pierwszego pola rekordu. Naciskając **CAPS SHIFT i 6 powodujemy przeskok kursora na początek następnego pola danych. W ten sposób zawartość wybranego pola może być poddana dowolnej korekcie. Po wprowadzeniu poprawek w danym pou na ekranie należy wcisnąć ENTER,** aby zmodyfikować rekord. Po przejściu kursora przez wszystkie pola program powraca do trybu zleceń.

INFORM — w obszarze nagłówka ekranu wyświetlane są podstawowe informacje: nazwa zbioru danych (nazwa, pod jaką zbiór był zapamiętany na taśmie), ilość pól wchodzących w skład jednego rekordu (**FIELDS**:), ilość bajtów pamięci wykorzystanych przez zbiór danych (**BYTES USED**:), ilość wolnych bajtów pamięci (**REMAINING**:) oraz aktualna liczba rekordów (RECORDS:) i numer pola, według którego uporządkowany jest cały zbiór danych (**ORDER FIELD**:). Powrót do trybu zleceń przez wciśnięcie dowolnego przycisku.

FORWARD — służy do wyświetlania kolejnego rekordu z całego lub zredukowanego za pomocą zlecenia **SELECT** zbioru danych.

BACK — analogicznie — wyświetlanie poprzedniego rekordu.

RESET — przejście do pierwszego rekordu zbioru danych.

ORDER — sortowanie zbioru danych według porządku leksykogra- ficznego względem dowolnie wybranego pola rekordu. Po zmianie nagłówka wciskając dowolny przycisk (oprócz **ENTER** powodujemy przemieszczenie kursora do następnego pola. Jeżeli wciśniemy **ENTER**, to program przejdzie do wyświetlania nagłówka ekranu ze zbiorem zleceń, a zbiór danych zostanie natychmiast uporządkowany względem pola, które wskazywał kursor.

SELECT — wyszukiwanie ze zbioru danych jego podzbiorów według zadanych kryteriów. Po wyborze zlecenia pojawi się nowy nagłówek. Jeśli wciśniemy teraz **ENTER**, oznaczać to będzie, że kryteria służące do wyszukiwania podzbiorów mają być stosowane do wszystkich pól. Jeśli jednak dowolnym innym przyciskiem przesuniemy kursor na początek któregoś z pól i naciśnie- my **ENTER**, to kryteria wyszukiwania będą stosowane tylko do tego pola. które zostało wskazane. W obu przypadkach pojawi się nowy nagłówek **SEARCH MODE**. Jeżeli teraz wprowadzimy dowolną sekwencję'znaków, to będzie ona stanowiła wzorzec do porównywania z zawartością zbioru. Wprowadzenie wzorca kończymy klawiszem **ENTER**. Jeśli podane warunki spełnia więcej niż jeden rekord, to pozostałe otrzymamy po ponownym wciśnięciu **ENTER**. W każdej chwili możemy wpisać nowy wzorzec.

Tryb przeszukiwania opuszczamy klawiszem **STOP**. Wówczas program żąda decyzji: czy ostatni wzorzec ma zostać aktywny, czy należy

go usunąć. Jeśli odpowiedź będzie twierdząca (Y), to zlecenia programu VU-FILE będą operowały na zredukowanym za pomocą wzorca zbiorze danych. Aktywny wzorzec wyświetlany jest poniżej zestawu zleceń. Jednokrotne naciśnięcie „S" spowoduje usunięcie ograniczeń.

QUIT — zakończenie pracy w trybie zleceń i przejście do ekranu ze zbiorem funkcji VU-FILE'a (szczegóły w dalszej części).

PRINT — służy do wydruku zbioru danych począwszy od aktualnie wyświetlanego rekordu. Kolejne drukowane rekordy wyświetlane są równocześnie na ekranie. Drukowanie można przerwać wciśnięciem dowolnego klawisza.

COPY — dokładne skopiowanie wyświetlanego na ekranie obrazu rekordu na drukarkę.

LIST — wyświetlanie kolejnych rekordów, począwszy od ostatnio wyświetlanego, w kolejności, w jakiej są uporządkowane, przez ok. 1 s każdy. Listowanie można przerwać wciskając dowolny klawisz.

DELETE — wymazanie ze zbioru aktualnie wyświetlanego rekordu. O decyzję program pyta ponownie. Należy wówczas nacisnąć **ENTER**.

UWAGA! Wszystkie wymienione powyżej zlecenia wywoływane są klawiszem z pierwszą literą ich nazwy!

Zlecenie **QUIT** powoduje przejście do zestawu funkcji bazy danych:

1 — rozpoczęcie pracy z VU-FILE (przejście do trybu zleceń, a więc to, co zostało opisane wyżej);
2 — zmiana formatu rekordu dla ekranu (pojawia się nagłówek (**RECORD LAYOUT...**) taki, jak na początku definiowania rekordu) — tym samym systemem można dokonać zmian i poprawek w dotychczasowym rozmieszczeniu tytułów pól i w samych polach);
3 — zmiana formatu rekordu dla wydruku w podobny sposób jak dla ekranu;
4 — zapis zbioru danych na taśmie;
5 — wczytanie nowego zbioru danych;
6 — usunięcie aktualnego zbioru danych.

Zapis odbywa się w sposób zbliżony do zapisu w systemie ZX Spectrum. Należy podać nazwę zbioru i po włączeniu magnetofonu nacisnąć dowolny klawisz. Klawiszem „V" można później zweryfikować zapis. W wypadku błędu w weryfikacji należy wpisać w trybie bezpośrednim **GO TO USR a.** Program powróci wtedy do zestawu funkcji VU-FILE. Po usunięciu danych można wpisać nowy zbiór z taśmy lub rozpocząć ponowne definiowanie rekordów.

VU-FILE nie jest programem nowym — powstał niemal natychmiast po ZX Spectrum, ale brak kontynuacji w tworzeniu nowych baz danych świadczy o tym, że tylko przy pomocy VU-FILE można efektywnie wykorzystać niewielkie przecież możliwości tego mikrokomputera. Trzeba tylko wiedzieć co i jak notować.

Piotr Bernatek

SPRÓBUJMY INACZEJ

Zdarza się często potrzeba wpisywania do pamięci komputera listingów programów w kodzie maszynowym. Listingi takie to liczby w liniach DATA w postaci cyfr dziesiętnych, lub tekstu znakowego przy zapisie heksadecynalnym. Zapis ten bardzo często jest dublowany dodatkowym programem w asemblerze.

Osobiście nie korzystam z żadnego z tych sposobów uważając je za mało efektywne przy wklepywaniu danych z następujących powodów:

1. Skoro mamy w posiadaniu program w bezpośrednim kodzie maszynowym, pisanie go ponownie w asemblerze i jego disasemblacja jest podobna do sceny faceta opalającego się nad wodą w kąpielówkach, który przed kąpielą ubiera się w smoking, by ponownie rozebrać się do kąpielówek przed zanurzeniem do wody (pomijam sprawy dydaktyczne, lub możliwości skorygowania błędnych czy nieczytelnych danych);
2. Formułowania danych w liniach DATA związane jest z koniecznością tworzenia tych linii, dodatkowo każdą liczbę należy rozdzielić przecinkiem. Wprawdzie nie jest to wymagane przy danych znakowych, lecz odczyt tekstu w linii jest w tym przypadku bardzo utrudniony i prowadzi do licznych pomyłek;
3. Wpisanie do pamięci danych z linii DATA wymaga napisania dodatkowego programu w języku BASIC, który (szczególnie przy danych tekstowych) jest dosyć złożony. Wymienione czynności są czasochłonne, co bardzo często skutecznie zniechęca do napisania i wykorzystania wielu ciekawych procedur w kodzie maszynowym.

Do wpisywania kodu maszynowego wykorzystuję uniwersalny, utrwalony na taśmie program, pozwalający na wpisywanie danych (o podanym adresie początkowym), jak również i kontrolny odczyt zawartości każdej komórki pamięci komputera. Taki mikroprogramik w przypadku „wypadnięcia" (co w programach KM nie jest zjawiskiem sporadycznym) jest bardzo łatwy do ponownego wprowadzenia do komputera. Zaś wpisywanie kodu maszynowego odbywa się bezpośrednio pod wskazane adresy 1 może być wykonane w czasie co najmniej dwukrotnie mniejszym od czasu potrzebnego w innych metodach.

Programik składa się z dwóch części:
I wpisującej — linie 2 i 3
II. odczytującej — linie 5 i 6 (linie 7, 8, 9 stanowią wspólny podprogram, tak dla części wpisującej, jak i odczytującej).

Uwagi dotyczące działania i obsługi programu:
1. Program tak przy wpisywaniu jak i odczycie po każdych 10 bajtach podaje sumę kontrolną modulo 1000 ich zawartości;
2. Przy pytaniu „Adres" lub „poz" wpisujemy adres 1 komórki KM;
3. W przypadku błędnego wpisu naciskany „P" (lub inną literę) i przez GO TO 1 (nie naciskamy ENTER) podajemy adres pierwszego bajtu danej dziesiątki w której wystąpił błąd;

Niżej przedstawiam podobny program narzędziowy posiadający następujące możliwości:
1. wpisywania i odczytywania jednobajtowych i dwubajtowych komórek pamięci Spectrum
2. interpretacje każdej komórki w języku BASIC
3. wydruk na ekranie adresu dowolnej linii, w której znajduje się kursor.

Program zawiera 256 bajtów kodu maszynowego i znajduje się w buforze drukarki od adresu 23296. Program uruchomiany jest instrukcją RANDOMIZE USR 23296.

Zaletą programu jest możliwość bezkolizyjnego analizowania innych programów znajdujących się powyżej adresu PROG.

Osobiście wykorzystuję go również do pisania niezbyt długich programów w kodzie maszynowym (program w kodzie maszynowym wpisuje się do komputera za pomocą poprzedniego programu niecałe 10 min. sprawdziłem).

Mieczysław Torbus

Od redakcji: Jest to oczywiście tylko propozycja. Decyzja należy do Czytelników — prosimy o uwagi.

```
2 POKE 23609,30: INPUT "
Adres: ";n: LET a=n: LET
s=0: REM wp is
3 INPUT " Liczba: ";x: IF
x>255 THEN BEEP .1,1: GO TO
3
4 POKE n,x: LET x=PEEK n: GO
SUB 7: GO TO 3
5 INPUT " poz. ";n: LET a=n
: LET s=0: REM Odczyt
6 LET x=PEEK n: GO SUB 7: GO
TO 5
7 LET s=s+x: IF S>999 THEN
LET s=s-1000
8 PRINT n; TAB 10;x : LET
n=n+1 : IF n-a=10 THEN PRINT
TAB 10;s: LET a=a+10: LET
S=0: PRINT
9 RETURN
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	195	110	92	34	31	198	176	205	91	2	205	8	92
1	188	42	201	176	31	91	92	89	205	205	148	205	111
2	91	75	62	92	31	34	115	91	89	1	91	148	96
3	0	92	42	62	31	85	40	205	91	22	255	91	62
4	62	119	215	247	56	92	8	89	205	42	229	225	16
5	30	33	0	219	243	201	35	91	94	176	86	126	205
6	50	228	205	254	205	205	114	24	91	92	43	245	148
7	57	91	160	31	172	160	205	227	24	229	94	35	91
8	92	34	91	48	91	91	172	205	251	62	235	34	62
9	62	85	235	38	33	42	91	172	62	1	62	176	21
	827	909	303	201	924	199	48	398	203	832	418	293	904
0	205	245	91	42	190	9	92	25	97	14	24	58	0
1	140	62	68	75	201	205	205	0	58	0	0	32	0
2	91	23	77	92	197	181	110	238	32	0	238	34	22
3	241	215	205	35	213	3	25	34	34	56	34	59	91
4	254	241	43	35	229	225	68	32	59	91	32	110	0
5	32	215	45	35	17	209	77	108	110	0	65	58	13
6	56	215	205	94	50	193	201	105	58	13	100	249	255
7	1	201	227	35	0	201	0	99	249	0	114	192	0
8	215	205	45	86	33	42	0	122	192	0	101	48	29
9	201	140	201	175	0	73	1	98	48	1	115	14	32
	436	762	207	704	130	341	779	861	937	175	823	854	442

OD ŚRODKA

cz. VII

Miesiąc temu przedstawiliśmy wam procedury z ROM-u — SAVE-BYTES i LOAD-BYTES. Dziś zobaczycie, jak procedury te wykorzystywane są w zabezpieczeniach programów. Zajmiemy się blokami kodu maszynowego, które uruchamiają się w dziwny sposób.

Pierwszym z tych sposobów jest przykrywanie programu ładującego blokiem, który jest przez niego ładowany. Zabezpieczenie takie występuje np. w grze „Three Weeks in Paradise". Prześledźmy sposób wczytywania tego programu, tak, by się nie uruchomił.

Zaczynamy jak zwykle od BASIC-u i przystępujemy do jego oglądania. Okazuje się, że praktycznie jedyną ważną instrukcją jest RANDOMIZE USR w obszar zmiennych BASIC-a, czyli tam znajduje się procedura ładująca, napisana w asemblerze. Musimy więc zająć się nią.

Najlepiej jest zdisasemblować tę procedurę począwszy od adresu, od którego jest uruchamiana instrukcją RANDOMIZE USR (PEEK 23627 + 256 PEEK 23628), tzn. w naszym przypadku, od adresu 24130. Procedury takie wykorzystują zwykle znaną nam procedurę 1366, wczytującą bloki bez nagłówka. Tak jest i tu, ale przed jej wywołaniem, za pomocą rozkazu LDIR, procedura ładująca przenosi samą siebie na koniec pamięci (pod adres 63116) i skacze tam instrukcją JP:

```
24130       DI              ;Zabronienia przerwan
24131       LD SP,0         ; - Tin. LD SP,63536.
24134       LD HL,(23627)   ;Do HL adres o 28
24137       LD DE, 28       ;wiekszy, niz wartosc
24140       ADD HL,DE       ;zmiennej syst. VARS.
24141       LD DE,63116     ;Do DE adres, a do BC
24144       LD BC,196       ;dtugosc bloku, ktory
24147       LDIR            ;zostan skopiowany.
24149       JP 63116        ;Kontynuacja wykony-
24152       LD IX,16384     ;-wania programu pod
24156       LD DE,6912      ;innym adresem
```

Teraz wczytuje obrazek, a następnie główny blok danych:

```
63116       LD IX,16384     ;Przygotowania do za-
63120       LD DE,6912      ;ladowania obrazka na
63123       LD A, 255       ;ekran, poprzez pro-;
63125       SCF             ;cedure LOAD-BYTES z
63126       CALL 1366       ;ROM-u.
63129       LD IX,26490     ;Parametry glownego
63133       LD DE,38582     ;bloku programu,ktory
63136       LD A,255        ;wczytujac sie,kasuje
63138       SCF             ;te procedure.
63139       CALL 1366       ;wczytanie teoo bloku
63142       JR NZ,+79       ;Po powrocie z proce-
63144       CP A            ;dury 1366 znajduje
63146       CALL 65191      ;sie tu juz inny pro-
63149       JR NC,-2        ;gram.
```

Wyjaśnienia wymaga jednak sposób, w jaki uruchamia się wczytany program. Jak zapewne wiecie, każda instrukcja CALL odkłada na stosie maszynowym adres, od którego ma działać program po powrocie z podprogramu. W tym loaderze, po wykonaniu drugiej instrukcji CALL 1366, na stosie odkładany jest adres instrukcji następnej po CALL, tzn. 63142, a procedura ładująca najzwyczajniej w świecie kasuje samą siebie, bo bajty z magnetofonu wczytują się w ten sam obszar pamięci, gdzie była ona umieszczona — wczytywany program „przykrywa" procedurę ładującą. Najistotniejszą rzeczą jest sposób, w jaki wczytany program się uruchamia: Procedura 1366 kończy się oczywiście instrukcją RET, która oznacza skok pod adres zapisany na stosie, czyli w naszym przypadku, pod adres 63142. W trakcie wczytania programu procedura, która się tam znajdowała, została podmieniona przez wczytany właśnie program, ale mikroprocesor nie zauważa tego — wraca pod adres, z którego wykonano CALL 1366, nie zwracając uwagi na to, że znajduje się tam już zupełnie inny program. Schematycznie przedstawia to rys.1. Z lewej strony rozpisana jest zawartość pamięci przed, a z prawej — po wczytaniu programu. Żółtym kolorem zaznaczone są instrukcje, składające się na wykonywany program.

A oto, jak rozpoznać tego typu zabezpieczenia, i jak ' je zlikwidować. Zaczynamy oczywiście od BASIC-a, odczytujemy procedurę ładującą (w asemblerze) i liczymy adresy końców wczytywanych bloków (dodając do adresu początku (rejestr IX długość bloku/rejestr DE)). Jeśli którykolwiek z bloków zachodzi na procedurę ładującą, oznacza to, że program wczytuje się i uruchamia właśnie w ten sposób.

Reszta jest już prosta. Wystarczy, w oparciu o dane bloków (adres, i długość), napisać króciutką procedurę ładującą interesujący nas blok kodu lub spreparować odpowiedni nagłówek, następnie przez CLEAR adr odpowiednio ustawić stos maszynowy (aby wczytany program nie zniszczył stosu) i wreszcie — wczytać program. Po wykonaniu w nim odpowiednich zmian, nagrywamy go na taśmę, ale tak samo, jak był on zapisany oryginalnie (zgadzać się musi przede wszystkim długość). Jeśli blok ten był bez nagłówka (a tak jest w naszym przypadku), to nagrywamy go przez zwykłe SAVE „..."CODE..., ale pomijając nagłówek, tzn. uruchamiamy magnetofon dopiero w przerwie między nagłówkiem, a blokiem kodu. Można także próbować uruchomić wczytany blok skacząc pod odpowiedni adres instrukcją RANDOMIZE USR..., lecz nie zawsze musi się to udać. W grze „Three Weeks in Paradise" adresem tym będzie oczywiście 63142 i jak możesz się przekonać — metoda ta skutkuje.

Innym ciekawym sposobem uruchamiania bloków kodu maszynowego jest wczytywanie programu w obszar stosu maszynowego. W ten sposób można uruchamiać bloki kodu maszynowego, ładując je po prostu przez LOAD „"CODE! Metoda ta została schematycznie przedstawiona na rys.2. Wskaźnik stosu (rejestr SP) przyjmuje pokazaną na rysunku wartość w trakcie wykonywania procedury 1366 (wywołanej z BASIC-a przez LOAD „ „CODE).

Sposób, w jaki program się uruchamia, jest w sumie bardzo prosty. Adres wczytywania bloku jest tak wyliczony, że wczytuje się on na stos maszynowy, dokładnie od tego miejsca, w którym znajduje się (zapisany przez interpreter BASIC-a) adres powrotu z intrukcji LOAD „"CODE (jest on wtedy równy wartości zmiennej systemowej ERRSP — 2) lub wręcz z procedury LOAD-BYTES (równy ERRSP — 6). Wówczas dwa pierwsze bajty programu oznaczają adres jego uruchomienia. Sposób ten jest bardzo podobny do poprzedniego, tylko że tam podmieniana była procedura ładująca, tutaj — adres powrotu z tej procedury lub po prostu adres powrotu z instrukcji LOAD. Po wczytaniu bloku kodu, mikrokomputer odczytuje zawartość stosu i skacze pod odczytany adres (który dopiero co znalazł się w pamięci wraz z wczytanym programem). Pod tym adresem znajduje się w programie początek procedury ładującej kolejne jego bloki — tak jak to widać na rys.2.

Metoda ominięcia tego zabezpieczenia także jest dosyć prosta. Wystarczy zmienić RAMTOP na odpowiednio niską wartość, a następnie wczytać blok kodu, który dzięki temu nie uruchomi się. Sytuacja komplikuje się, jeśli blok ten jest bardzo długi (co zdarza się rzadko, ale jest możliwe) — wtedy musimy z nim postępować tak, jak z każdym zbyt długim blokiem, ale pamiętając, w jaki sposób się uruchamiał, (tzn. od jakiego adresu).

Zajmiemy się teraz właśnie cięciem bloków kodu o długości przekraczającej 42K. Włamywanie się do tego typu bloków polega na podzieleniu ich na takie fragmenty, by w pamięci pozostało jeszcze miejsce na MONS-a czy inny disasembler, poprawieniu tych frag-

mentów, a następnie „sklejeniu" ich w jedną całość lub napisaniu nowej procedury ładującej. Zazwyczaj wystarcza, jeśli długi blok podzielimy na dwie części. Aby otrzymać pierwszą z nich, wykorzystujemy procedurę 1366 ale z innymi parametrami niż wymaga tego dzielony na części blok (wcześniej oczywiście z procedury ładującej, lub jeśli takiej nie ma — z nagłówka tego bloku, odczytujemy jego długość i adres wczytania). Podajemy po prostu adres, pod jaki chcemy ten blok wczytać (powyżej RAMTOP-u!) oraz długość równą około 16K (mimo tego. że blok ten jest znacznie dłuższy). Wczytujemy teraz ten blok przez CA'LL 1366 lub CALL 2050, lecz w drugim przypadku komunikat „Tape loading error", który się ponowi, nie da nam żadnej informacji o poprawności wczytania — ładujemy tylko część bloku, a więc bez bajtu kontrolnego, który umieszczony jest na końcu. Tak wczytaną pierwszą część bloku nagrywamy na taśmę a na razie zabieramy się do drugiej części. Jej wczytanie jest trudniejsze, ale też możliwe, mimo ograniczeń pojemności pamięci. Wystarczy wykorzystać fakt, że w Spectrum istnieje 16K ROM-u, a próba zapisu informacji do ROM-u po prostu nic nie daje. Jeśli więc np. wywołamy procedurę 1366 z adresem wczytania równym np. O, to początkowe 16K wczytywanego bloku zostanie stracone, a do pamięci RAM wczyta się TYLKO końcowe 32K lub mniej (zależnie od długości bloku). Nie wystarczy jednak napisanie tak prostej procedurki, jak poprzednio. Wczytywany blok zajmie w pamięci RAM adresy od 16384 i dalej, zachodząc na zmienne systemowe, pozostawiając bez zmian jedynie bajty o adresach większych niż jego długość. Dlatego też musimy zadbać o to, by stos maszynowy oraz napisaną przez nas procedurę ładującą umieścić na końcu pamięci. Musimy także pamiętać o tym, że system BASIC-a zostaje zniszczony i musimy wczytany blok od razu nagrać na taśmę procedurę napisaną w asemblerze. Ponadto w czasie między wczytaniem fragmentu bloku a jego nagraniem, nie można do odblokowania przerwań, gdyż mnieniają one wartość komórek o adresach 23552 — 23560, oraz 23672 — 23673, a tam znajduje się wczytany blok. Aby spełnić ten ostatni warunek, wskoczymy do środka procedury 1366, dzięki czemu po wczytaniu bloku nie zostanie wykonana procedura 1343, która właśnie m.in. odblokowuje przerwania. Przez CLEAR 64999 przenosimy stos maszynowy, a od adresu 65500 umieszczamy procedurę ładującą:

```
 10         ORG 65000
 20         LD IX,0         ;Adres wczytania.
 30         LD DE,dl        ;Dlugosc bloku.
 40         LD A,255        ;Przygotowania do
 50         SCF             ;wczytania bloku.
 60         INC D           ;W tam sposob z a-
 70         EX AF.AF'       ;stepujemy pocza-
 80         DEC D           ;tek procedury
 90         DI              ;1366, a nastep-
100         LD A,15         ;nie wskakujmy
110         OUT (254),A     ;do jej wnetrza:
120         CALL 1378
130         LD A,0          ;Czarna ranka oz-
140         JR C,OK         ;naczac bedzie po-
150         LD A, 7         ;prawne wczytanie,
160 OK      OUT (254),A     ;biala - biedne.
170 CZEKAJ  LD A,191        ;Czekamy na wcis-
180         IN A,(254)      ;nieci* klawisza
190         RRA             ;ENTER.
```

listing 1

```
10 CLEAR 64999
20 INPUT "dlugosc bloku
   ?";dl:RANDOMIZE dl: LET x=PEEK
   23670: LET y=PEEK 23671: LET s=0
30 FOR n=65000 TO 65053: READ a:
   LET s=s+a: POKE n,a: NEXT n
40 IF s<>5254+2*(x+y)-64 THEN
   PRINT "Blad w danych!": STOP
50 PRINT "Wszystkie dane
   Ok."'"Wlacz magnetofon"
60 RANDOMIZE USR 65500
70 DATA 221,33,0,0,17,x,y,62,255,
   53,20,8,21,243,62,15,211,254
80 DATA 205,98,5,62,0,56,2,62,7,
   211,254,62,191,219,254,31,56
90 DATA 249,221,33,0,0,17,x, y-64,62,
   255,205,194,4,33,231,253,202,25,18
```

```
200        JR C,CZEKAJ
210        LD IX,0          ;Nagrywamy wczy-
220        LD DE,dl-16384   ;tany blok kodu
230        LD A, 255        ;na tasme.
240        CALL 1218        ;oraz inicjujemy
230        LD HL,64999      ;dzialania sys-
260        JP 4633          ;mu BASIC-a.
```

Zamiast wczytywać asembler i .wpisywać powyższą procedurę, możesz uruchomić program z listingu 1.

Jak wygląda cięcie bloku? Ustawiamy taśmę na bloku kodu, który chcemy ciąć. Jeśli miał on nagłówek — to pomijamy go. Uruchamiamy procedurę i włączamy magnetofon. Nie przestrasz się, gdy w pewnym momencie ujrzysz, że program wczytuje się na ekran — tak przecież powinno być. Po załadowaniu bloku kolor ramki sygnalizuje poprawność wczytania: jeśli ramka jest czarna, to wszystko jest OK., jeśli biała — wystąpił błąd. Teraz włóż do magnetofonu drugą kasetę, włącz nagrywanie i wciśnij ENTER. Program nagra się na taśmę, a potem procedura wróci do BASIC-a, inicjując system komunikatem „ 1982...", ale nie kasując pla- mięci (niszczony jest jedynie obszar od początku ekranu, do adresu ok. 24000). Teraz, preparując nagłówek lub pisząc króciutką procedurkę, możesz wczytać uzyskany blok pod dowolny adres.

Gdy już znajdziesz to, czego szukałeś, i chcesz uruchomić poprawiony program, musisz się trochę pomęczyć i „skleić" „rozcięty" program lub napisać do niego nową procedurę ładującą. Jeżeli program wypełniał całe 48K pamięci RAM, możliwa jest tylko ta druga metoda.

Jeśli chcesz skleić bloki — wystarczy napisać procedurę podobną do rozcinającej, lecz która wczyta pierwszy blok pod adres 16384, drugi — zaraz za nim, a następnie nagra je razem, jako jeden blok.

W następnym, ostatnim już, odcinku naszego cyklu — odkodowywanie zakodowanych bloków „Bytes".

Robert Dudzik
Tomasz Surmacz

Od red. W czwartym odcinku naszego cyklu zamienione zostały dwa listingi w tekście — przepraszamy.

Rys.1.
Program ładujący
Wczytany program
LD A,255
SCF
CALL 1366
JR NZ,+79
CP A
CALL 65191
...
63136 (254)
63137 (254)
63138 (75)
63139 (254)
63140 (255)
63141 (255)
... DI
LOAD-BYTES 1366
LD SP,0
...
63146 LD HL,
63147 28267
...

Rys.2.
Stos maszynowy
...
65356 ...
rej. SP→
65359 1343
65360 2053
65361
65362 7030 65368
65363
65364 4867 ?
65365
... ? ?
RAMTOP→ 62 62
65368 LD IX,
65369 ...
65370
... LD DE,

MINI WYCIECZKA

W ŚWIAT BAZ DANYCH

Motto:
"Przy samych danych bez baz bałagan byłby w sam raz..."

Wprawdzie w dzisiejszych czasach informacja jest cennym towarem, jednakże samo gromadzenie i posiadanie informacji w pamięci komputera bez możliwości szybkiego wyszukiwania konkretnych danych nie miałoby większego sensu.

Konieczne stało się więc opracowanie specjalnych programów ułatwiających przechowywanie, modyfikację, uaktualnianie i wyszukiwanie informacji. I w ten sposób pojawiły się systemy zarządzania bazą danych, w których informacje wprowadzane i zapamiętywane są w postaci zorganizowanych struktur: pól, rekordów i plików. Podstawową strukturą jest rekord, który może być podzielony na pola, zaś pewna ilość rekordów (różna w różnych bazach danych) tworzy plik. Jeżeli wyobrazimy sobie rekord jako pewien formularz to jego rubryki będą polami, a paczka formularzy będzie plikiem. W zależności od powiązań między strukturami baz danych i organizacją samych struktur ustalił się podział na: kartotekowe bazy danych, relacyjne bazy danych, hierarchiczne, sieciowe, bazy danych o formacie swobodnym i wielodostępne. Ograniczę się do omówienia tylko dwóch pierwszych typów, do których istnieją stosowne programy pracujące na mikrokomputerach Amstrada.

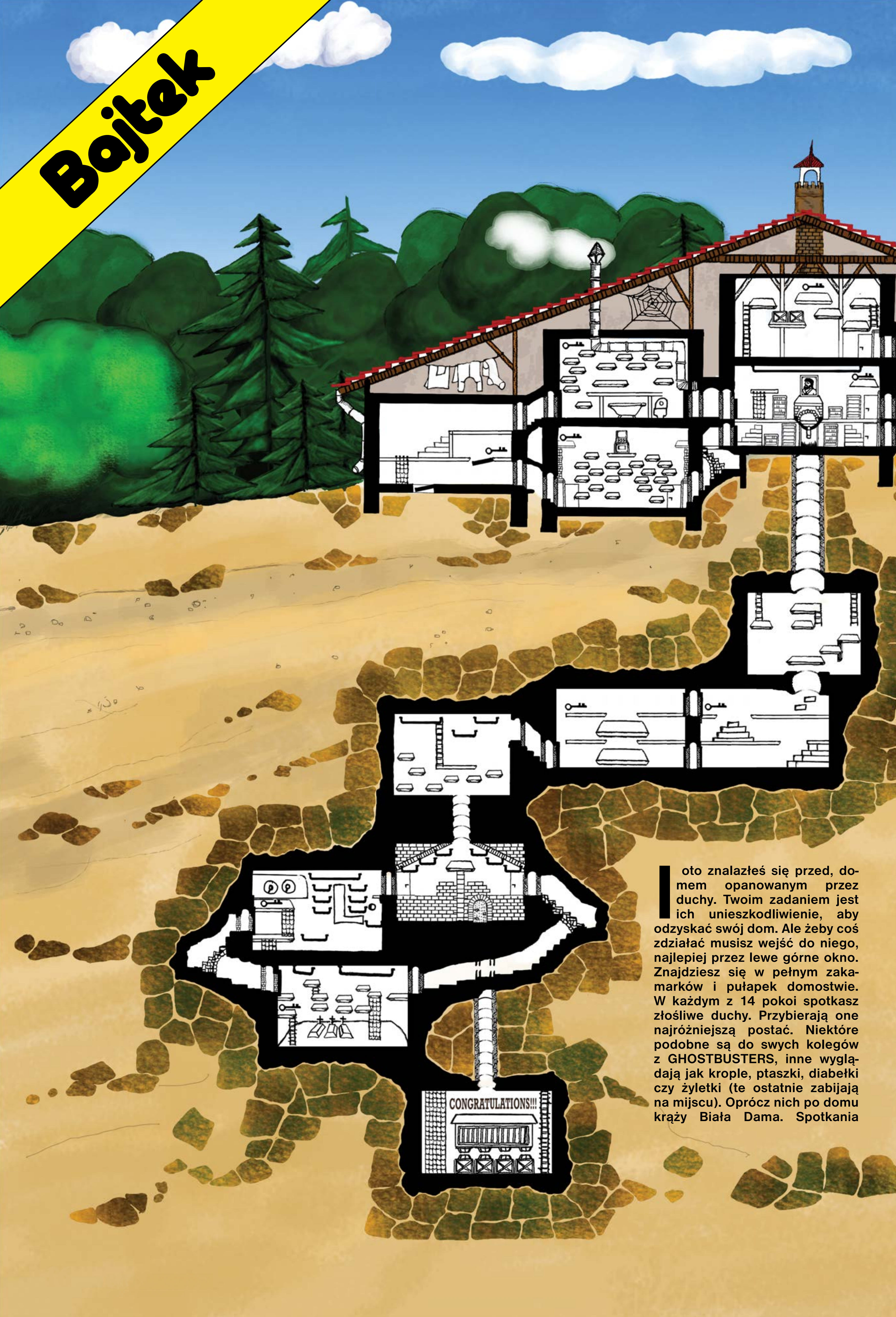
1. Kartotekowe bazy danych. Plik, będący magazynem naszych informacji, składa się z rekordów o takiej samej strukturze, tzn. z pól o tych samych nazwach i tego samego typu, aczkolwiek treść informacji wpisana w te pola może być dla każdego z rekordów inna. Wróćmy do przykładu formularza (rekord) rubrykami (pola): wiek, płeć, miejsce zamieszkania... Jeżeli taki formularz wypełnia mężczyzna i kobieta w różnym wieku i z różnych miast, to otrzymamy prosty plik (dwa formularze) z różnymi informacjami w rekordach o takich samych polach.

W tej grupie programów wyróżnia się prostotą obsługi Gestion de Fichiers — baza danych, w której każdy rekord może mieć do 19 pól o szerokości 50 znaków. Wielkość pliku ograniczona jest jedynie pojemnością dyskietki, jednakże użytkownik „obarczony" jest koniecznością nadania nazw rekordom, określenia ilości i nazw pól i, oczywiście, wypełnienia ich treścią. Taka elastyczność jest dużą zaletą dla pracowitych użytkowników mikrokomputerów. Leniwi mogą korzystać ze specjalizowanych prostszych baz danych typu „notes" (Agendafirmy Logys), czy „książka adresowa" (Carnet d'adresse tej samej firmy). Wpisane wcześniej informacje mogą być wyszukiwane od kątemróżych kryteriów: nazwiska, adresu, numeru telefonu lub daty.

2. Relacyjne bazy danych. Pliki, tak jak w bazach kartotekowych zbudowane są z rekordów, a rekordy z pól. Jednakże baza może zawierać kilka plików, różniących się strukturą rekordów a także umożliwia tworzenie nowych rekordów z różnych plików i tworzenie nowej bazy danych zawierającej łączne informacje, pod warunkiem że przynajmniej jedno pole jest wspólne. Wyobraźmy sobie autora książek wydawanych w różnych wydawnictwach, który chce podsumować swój dorobek: każde wydawnictwo ma kartotekę wydanych własnych książek i tylko nazwisko zainteresowanego autora jest wspólne. Przy pomocy relacyjnej bazy danych, jeżeli wydawnictwa są połączone w sieci komputerowej, autor uzyska niemal naatychmiast potrzebne informacje, nawet nie wiedząc, że możliwe było utworzenie nowego rekordu z wielu innych, których jednym wspólnym polem było jego nazwisko. Najpopularniejszym programem w tej drupie jest niewątpliwie dBase II opracowany przez formę Ashton Tate — w pełni profesjonalny, pracujący początkowo pod kontrolą systemu operacyjnego CP/M, adaptowany później do mikrokomputerów 16-bitowych (PC, DOS, CP/M 80 i 86). W interesującej nas wersji można założyć bazę danych z maksymalną ilością 65535 rekordów, składających się z 32 pól. Każde pole może zawierać do 254 znaków, jednak łączna liczba znaków w rekordzie nie może przekroczyć 1000. Informacje w polach mogą być znakowe, numeryczne lub logiczne. Wbudowany w dBase II język programowania umożliwia, w przypadku wykonywania skomplikowanych operacji na polach i rekordach nieprzewidzianych w menu, stworzenie przez programistę lub zaawansowanego użytkownika swoistego programu użytkowego, ułatwiającego pracę innym użytkownikom tej bazy danych. W prostszych przypadkach wystarcza korzystanie z dołączonego do programu menu i wprowadzanie z klawiatury podstawowych poleceń oferowanych przez dBase II (np. zmiana struktury i zawartości rekordu, sortowanie pliku wg określonych kryteriów, dołączanie rekorsu do pliku, sumowanie wartości numerycznych ze wskazaniem rekordów, wyświetlanie lub wydruk danych, usuwanie rekordów spełniających zdefiniowany wcześniej warunek itp.). Istnieją również programy umożliwiające tworzenie mniejszych, ale łatwiejszych w obsłudze baz danych jak np. Datamat (do 400 rekordów) czy Masterdile — firmowa baza danych Amstrada (do 500 rekordów). Niezależnie do swoich zalet bazy danych są tylko pewnym Zbiorem informacji, który w świetle potrzeb użytkowników wymaga łączenia z tekstem, dokonywania różnorodnych obliczeń i prezentowania (łącznie lub wyłącznie) w postaci różnorodnych tabel i wykresów. Powstały więc Pakiety Zintegrowane, łączące edytory tekstów, bazy danych, programy kalkulacyjne, graficzne i komunikacyjne, z ujednoliconym sposobem obsługi wszystkich składników pakietu i łatwością przesyłania informacji między nimi. W przypadku CPC również Masterfile współpracuje z arkuszem kalkulacyjnym Mastercalc i edytorami tekstów Tasword oraz Protext. Podobnym minipakietem jest Pockte Base + Packet Cale. Dość duża uniwersalność pakietów zintegrowanych niesie za sobą również pewne ograniczenia, ale to już temat na zupełnie inne opowiadanie. Tymczasem proponuję rzucić okiem na krótki program będący przykładem kartotekowej bazy danych. Jest to notatnik, umożliwiający wpisywanie modyfikację i przechowywanie na dyskietce nazwisk, adresów i telefonów. Po uruchomieniu programu należy oczywiście najpierw zapisać notatnik (wybieramy opcję 1 z menu) i wpisać te informacje na dyskietkę (opcja 3) do tworzonego automatycznie zbioru „notki". Przy następnym korzystaniu z programu należy wybrać opcję 4 (odczyt danych z dyskietki) i już można poszukiwać informacji przez wybór opcji. 2. Program umożliwia również modyfikację i usuwanie poszczególnych zapisów, jednak należy pamiętać aby po każdej z tych czynności zapisać uaktualniony notatnik na dyskietkę (ponownie opcja 3).

Wojciech Ziółek

listing na str. 21



I oto znalazłeś się przed, domem opanowanym przez duchy. Twoim zadaniem jest ich unieszkodliwienie, aby odzyskać swój dom. Ale żeby coś zdziałać musisz wejść do niego, najlepiej przez lewe górne okno. Znajdziesz się w pełnym zakamarków i pułapek domostwie. W każdym z 14 pokoi spotkasz złośliwe duchy. Przybierają one najróżniejszą postać. Niektóre podobne są do swych kolegów z GHOSTBUSTERS, inne wyglądają jak krople, ptaszki, diabełki czy żyletki (te ostatnie zabijają na mijscu). Oprócz nich po domu krąży Biała Dama. Spotkania



ghost chaser

- — poręcz
- — zapadnia
- — schody
- — mata
- — drabina
- — drzwi otwarte
- — drzwi zamknięte
- — klucz
- — drzwi winda
- — obraz - wejście w podziemie

z nią możesz uniknąć rzucając w nią kamieniem podniesionym z podłogi.

Jeżeli zetkniesz się z duszkiem, przestraszysz się i stracisz trochę energii życiowej. Nie jest to bardzo niebezpieczne, ale uważaj, bo kilka takich zetknięć może cię kosztować życie. Powstaje kłopot, gdy spotkasz ducha będąc na poręczy. Spadasz wtedy na dół i zostaje po tobie mokra plama. Zawsze jednak możesz podkurczyć nogi.

W poruszaniu się po pokojach pomocne będą maty, schody i drabiny lecz uważaj na zapadnie i dziury w podłodze. Szczególnie niebezpieczna jest dziura w 11-tej komnacie.

Większość drzwi jest zamknięta — do każdych pasuje jeden klucz. W zasadzie zadanie sprowadza się do zebrania wszystkich kluczy. Nie zawsze jest to łatwe. Na pewno przyda ci się kilka wskazówek: Aby wziąć klucz z drugiej komnaty, trzeba wejść do trzeciej, zejść po schodach i macie na najniższy poziom i przejść teraz znów do drugiej komnaty. Uwaga na zapadnię! W czwartej komnacie możesz podskoczyć na skrzyni. W piątej pomoże ci szafa — winda ale uważaj, bo po schodach można wchodzić tylko w górę.

Po wzięciu kluczy z komnat 6,7 i 8 wejdziesz znowu do piątej i tu czeka cię niespodzianka. Otworzy się obrazek i ukazując tajne wejście do piwnic. Tu pozbieraj najpierw klucze z komnat 10 i 11, potem wróć do komnaty 10, zejdź na dół i idź dalej. W komnacie 12 uważaj na ptaszki.

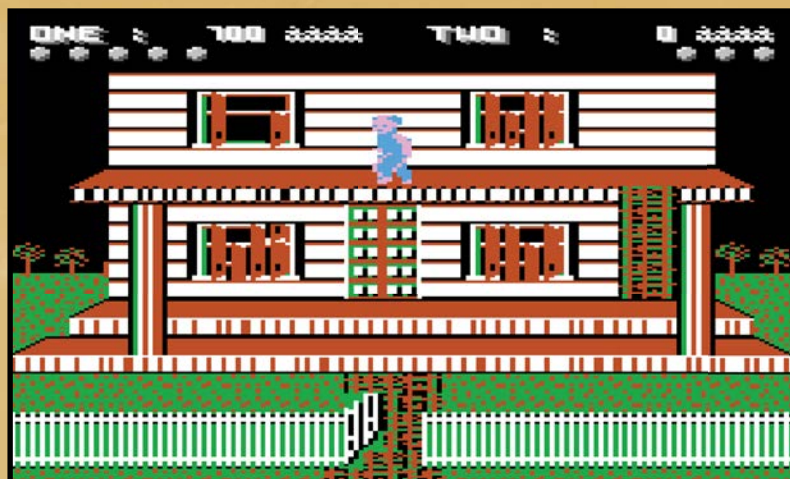
W 13 zejdź po poręczach i zabierz klucze z komnat 14 i 15. Teraz możesz zejść na dół po drabinie i wejść w drzwi po prawej stronie. Znajdziesz się w ostatniej komnacie. Tutaj możesz już tylko skakać z radości.

Na koniec dwie ważne informacje: wystukanie na klawiaturze słowa FANDA przenosi cię od razu do dziewiątej komnaty; FRANK odnawia zapas ludzików i daje na pewien czas nieśmiertelność.

Wchodząc do domu masz tylko jeden plan, trochę wiadomości i chęć wygranej. Nie jest to wiele ale warto spróbować, więc spróbuj!

Autor: Frank Cohen
Producent: Artwox Software
Komputer: Atari XL/XE

Robert Rogowski



10 BAJTKOWA LISTA PRZEBOJÓW (7/88)

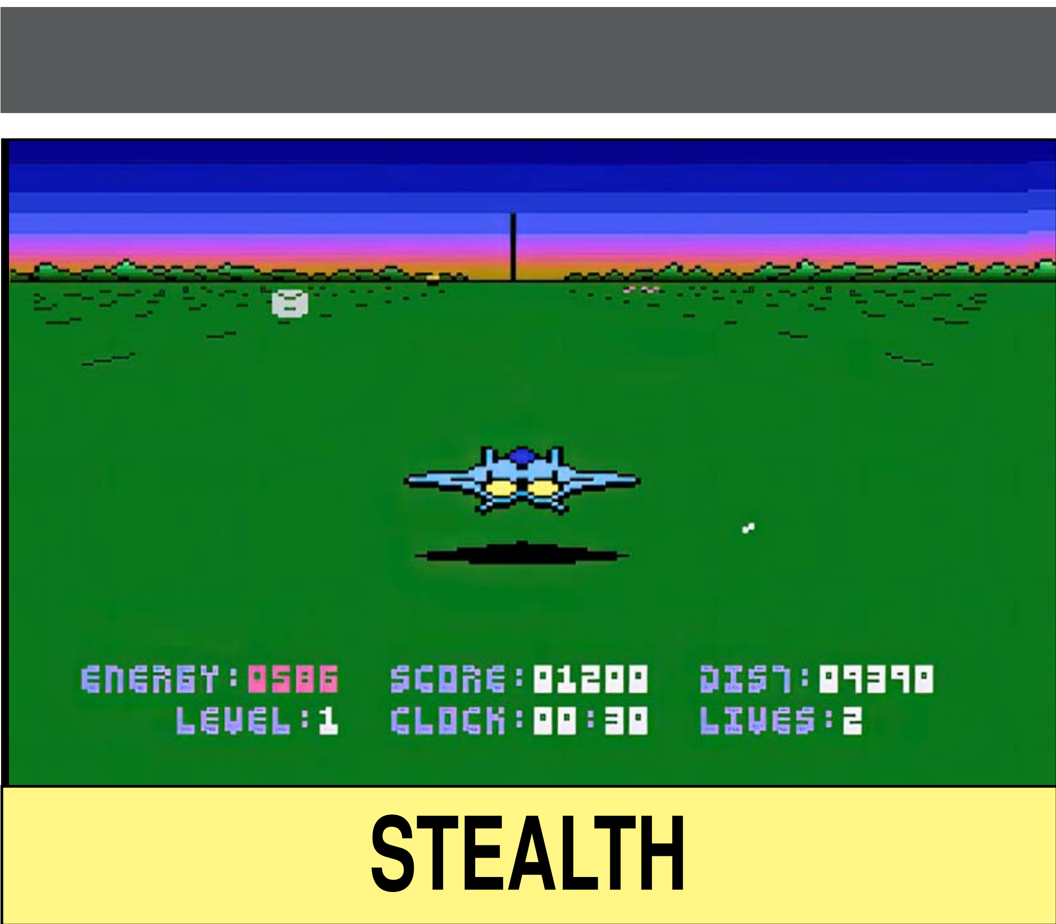
Nasza lista nadal bez większych rewelacji. To trochę dziwne, że Turbo Esprit, jedna z lepszych gier symulacyjnych, dopiero teraz wywalczyła pierwsze miejsce. Zaś Solo Flight, w niczym mu nie ustępujący, trzyma się dwie pozycje niżej. Jeszcze niżej jest trzeci symulator — Ace, który, jak narazie, dostępny jest tylko posiadaczom Spectrum i Commodore. Z pewnością jednak i on doczeka się wersji na Atari. Tym razem otrzymaliśmy 1273 propozycje na 132 tytuły.

			ATARI	AMSTRAD	COMMODORE	SPECTRUM
1	TURBO ESPRIT	!		x	x	x
2	SPLITTING IMAGES	!	x		x	
3	SOLO FLIGHT	↑		x	x	x
4	WEST BANK	↓		x	x	x
5	GHOST CHASER	↓	x		x	x
6	HARD BALL	↑	x	x	x	x
7	ACE	↓	x	x	x	x
8	SHOW JUMPING	↓		x		x
9	STAR QUAKE	↓		x	x	x
10	SABOTEUR II	!		x	x	x

Nagrody otrzymują: Tamara Pietkiewicz i Wojtek Grunert



Kopertę nadesłał Adam Karol z Płocka



STEALTH

Lewą ręką sprawdził zapięcie pasów bezpieczeństwa i nasunął na twarz pancerną szybę kasku. Ogniki odległego słońca zatańczyły na jego niklowanej powierzchni. Prawą, uzbrojoną w ciężką rękawicę bojową, ujął czarny drążek steru. Leżał doskonale w dłoni. Lampki kontrolne dwukolorową mozaiką oświetliły pulpit sterowniczy. Nie myślał o niczym. Za chwilę jego nerwy splotą się w jedną całość z elektronicznymi obwodami supermyśliwca. Lata treningu były poświęcone tej jednej chwili. Jego twarz stężała, a oczy nabrały drapieżnego blasku. W prawym, dolnym rogu ekranu zamigotała liczba 10 000 — dystans dzielący go od Wieży Ciemności. Kątem oka sprawdził jeszcze poziom paliwa i już miał pewność. Był gotowy do akcji.

Łagodnie zszedłeś nad powierzchnię planety wroga. Zimne dysze rozbłysły światłem silników fotonowych. Myśliwiec zaczął nabierać prędkości. Ziemia zdaje się umykać sprzed dziobu statku. Daleko na horyzoncie wystrzelona w tęczowe niebo widnieje Wieża Ciemności. Wyraźnie króluje nad ośnieżonymi łańcuchami odległych gór.

Nagły strzał z boku przywraca cię do rzeczywistości. Nie jesteś tu po to, aby podziwiać krajobrazy. Łagodny przechył myśliwca, lekkie dotknięcie spustu i z atakującego cię bunkra pozostaje tylko rozpraszający się z wolna obłok. Pociski wroga nie są w stanie naruszyć superpancerza twego myśliwca. Mogą jednak uszczuplić i tak skromne zapasy paliwa. Nie obawiaj się jednak — na swojej drodze spotkasz wiele pól energetycznych. Najlepiej wtedy zwolnić i poczekać aż basowe buczenie potwierdzi tankowanie. Korzystaj tylko z jasnych pól — fioletowe pozbawiają cię energii. Pamiętaj, że twój bak pomieści tylko 3000 litrów paliwa.

Siejąc po drodze zniszczenie dotarłeś wreszcie do Wieży Ciemności. Kilka pocisków i wieża wali się na ziemię. Twój myśliwiec gwałtownie przyspiesza i nabierając wysokości znika w oddali. Lecz twoja rola się nie kończy. Otrzymujesz awans do następnego etapu. Z każdym mnożą się niebezpieczeństwa — bunkry, czołgi, aktywne i wygasłe wulkany, samobójcze myśliwce. Tych ostatnich nie da się zestrzelić — ratuje tylko szybki uskok w bok. Pojawiają się w 3 etapie i towarzyszą do końca. Zaskakują przy tankowaniu paliwa. A pól energetycznych jest coraz mniej i nie można pozwolić sobie na opuszczenie żadnego. Czeka na ciebie wiele niespodzianek, których nie wolno mi wyjawić. Nie zdradzę nawet, ile etapów ma gra. Atrakcyjność „Stealth'a" bazuje bowiem na zaskoczeniu i tajemniczości.

Gra ta nie wymaga myślenia, ale za to posiada efektowną grafikę. Jest odprężeniem od trudnych programów symulacyjnych i skomplikowanych gier strategicznych.

Więc powodzenia, pilocie! W twoich rękach spoczywa los Galaktyki. Pamiętaj tylko o energii, nalotach myśliwców i tym, że 3 statki to wcale nie za dużo na pokonanie dystansu. Połamania skrzydeł i joysticka!

Komputer: Atari XL/XE, Commodore 64/128

(guma)

KRÓL I KRÓLOWA GIER



TAMARA PIETKIEWICZ z Sopotu, uczennica 4 klasy Szkoły Podstawowej nr 6. Posiada mikrokomputer Commodore 16. Ulubiona gra Rickatac, zainteresowania: tenis, jeździectwo, język niemiecki.



WOJTEK GRUNERT, uczeń Szkoły Podstawowej nr 27 w Gdańsku. Posiada mikrokomputer Timex. Ulubiona gra Uridiun oraz Basketball. Zainteresowania: matematyka, judo, piłka nożna, język angielski.



SOLO FLIGHT

SOLO FLIGHT(lot w pojedynkę) jest całkiem niezłym symulatorem lotu. Pozwoli Ci zaznać wielu trudów i niebezpieczeństw lotu prawdziwym samolotem. Do zapoznania się z samolotem i jego reakcjami służy opcja FLYING. W czasie treningów postaraj się dobrze opanować start, lądowanie, nawigację, lot na przyrządy (bardzo pomaga przy niskim pułapie chmur) oraz postępowanie w czasie zagrożeń i awarii.

Jeśli już dobrze opanowałeś wszystkie te czynności, możesz wybrać opcję MAIL PILOT (pilot pocztowy). Tu musisz się wykazać wszystkimi zdobytymi wcześniej umiejętnościami, gdyż punkty dostaniesz nie tylko za doręczenie przesyłki, ale i za punktualność. Możesz też wybrać, nad którym ze stanów USA chcesz latać (do wyboru masz Kansas, Waszyngton i Kolorado). Oprócz tego możesz wybrać stopień trudności gry lotu terenowego: **CLEAR** — ładna pogoda, **LANDING** — trening lądowania, **WINDY** — pogoda wietrzna lub IFR — lot na przyrządy przy niskim pułapie chmur, lub jeden ze stopni trudności lotu pocztowego: **STUDENT, PRIVATE, SENIOR lub COMMAND.**

W locie pomocne Ci będą przyrządy kontrolne i namiarowe, umieszczone w dolnej połowie ekranu. Duża tarcza z podziałką z lewej strony tablicy z podpisem **ALT** (altitude) to Twój wysokościomierz. Każdy znak na podziałce oznacza 1000 stóp dla małej wskazówki. Tarcza z oznaczeniem **SPD** (Speed) to prędkościomierz. Podaje on prędkość w węzłach od 0 do 180. Wskaźnik z podpisem **POWER** (moc) informuje o aktualnej mocy silników. Mała tarcza umieszczona w centrum to sztuczny horyzont, który pokazuje położenie samolotu względem prawdziwego horyzontu. Oprócz tych wskaźników, na dole po lewej stronie znajdują się cztery wskaźniki cyfrowe. Są to: **PITH** pokazujący nachylenie samolotu względem poziomu, **HEAD** pokazujący azymut (kąt między północą a kierunkiem lotu), **FLAPS** pokazujący kąt wychylenia klap oraz **CLIMB**, który jest wskaźnikiem prędkości pionowej. Pasek podpisany **FUEL** informuje o zawartości baku. Lampka z podpi sem TEMP zapala się, gdy silnik ulega przegrzaniu. Lampka z literką **L** pokazuje, czy zostało wypuszczone podwozie, a lampka z literką **B** oznacza włączone hamulce kół. Przyrządy nawigacyjne umieszczone są na prawej stronie tablicy. Są to **VOR1** i **VOR2**, które odczytują namiary ze stacji **VOR** (Very High Frequency Omnidirectional Radio Range — radiolatarnia ogólnokierunkowa o dużej częstotliwości) oraz wskaźnik **ILS** pomocny w trakcie lądowania.

W prawdziwych samolotach do latania używa się głównie sterów pionowych, poziomych oraz przepustnicy (regulacji mocy silników). Aby używać sterów, wystarczy Ci joystick (steru pionowego możesz używać po wciśnięciu **FIRE**. Do regulacji mocy potrzebne Ci będą klawisze od 0 do 9. (0 to wyłączone silniki, 9 to pełna moc). Do wypuszczania i chowania podwozia służy klawisz **L**. Klawisz F reguluje kątem wychylenia klap, a klawisz **B** uruchamia i zwalnia hamulce. Wciskając klawisze kursorów będziesz mógł zobaczyć, co dzieje się z tyłu i po bokach. Jeśli się zmęczysz, możesz wcisnąć klawisz P. Jeśli chcesz obniżyć wysokość lotu bez utraty kursu, przechyl samolot na skrzydło i wciśnij **FIRE** wykonasz tzw. ślizg pod wiatr.

Jeśli znudził Ci się sielankowy lot wciśnij klawisz **E**. Komputer zrobi wtedy specjalnie dla Ciebie jakąś awarię np. wysokościomierza.

W górnej części ekranu pokazany jest Twój samolot oraz cechy charakterystyczne terenu, nad którym się znajdujesz. Jeśli lecisz na małej wysokości, to na ziemi widoczny jest także cień Twojego samolotu. Lotniska są zaznaczone na czarno, farmy na szaro, wieże VOR i góry na biało.

Jeśli już opanowałeś podstawy lotu, wiesz gdzie jest jaki klawisz, możesz zacząć grać w pilota pocztowego. Wybierz opcję MAIL PILOT. Na ekranie ukaże się mapa (tylko w wersji dyskowej). Wciśnij teraz **START** — będziesz mógł załadować przesyłki pocztowe (przy pomocy klawisza **OPTION**). Możesz ich wziąć ile chcesz, ale pamiętaj, że każdy worek zwiększa masę Twego samolotu. Klawiszem **SELECT** możesz tankować paliwo (zbiornik ma pojemność 64 galony).

Jeśli wziąłeś już wszystko, co trzeba (zdrowy rozsądek też), wciśnij **START**. Na ekranie ponownie pojawi się mapa. Przestudiuj ją dokładnie, gdyż informacje o położeniu charakterystycznych punktów, gór i lotnisk mogą się okazać niezbędne. Wciśnij znowu **START**. Zaczyna się Twój lot.

Po przybyciu do celu wciśnij **START**. Ukaże się obraz poczty. Tutaj wszystko się powtarza i jest tak do momentu aż doręczysz pięć przesyłek lub rozbijesz się.

Pamiętaj, by nigdy do końca nie wierzyć przyrządom. Jeśli uważasz, że masz awarię, nie wpadaj w panikę. Postaraj się spokojnie wylądować na jakimś lotnisku, a Twoimi awariami zajmie się wyspecjalizowany personel naprawczy.

Najtrudniejszą umiejętnością do opanowania jest lądowanie. Zbliżając się do lotniska docelowego postaraj się ustawić samolot na przedłużeniu pasa startowego. Leć mniej więcej na wysokości 1000 stóp nad poziomem ziemi (Twój wysokościomierz pokazuje wysokość nad poziomem morza!). Gdy tylko uaktywni się wskaźnik ILS (mały punkt w środku tarczy) wtedy wznieś się lub opuść tak, by punkt był w środku tarczy. Wypuść podwozie, zmniejsz moc do ok. 4 i ustaw samolot tak by prędkość wznoszenia była ok. -4 (wskaźnik **CLIMB**) stopy na minutę. Gdy dostatecznie zbliżysz się do pasa startowego, zredukuj tempo obniżania. Teraz możesz już lądować na pasie, ale uważaj by prędkość samolotu nie była zbyt duża, gdyż wtedy samolot przeleci za daleko i nie zdążysz wyhamować. Prędkość nie może być też zbyt mała, wtedy samolot spadnie na ziemię, zanim doleci do pasa. Gdy już miękko osiądziesz na ziemi, wyłącz silniki (klawisz 0), uruchom hamulce i postaraj się wyhamować przed końcem pasa.

Jeśli lądowanie nie uda się, na ekranie ukaże się informacja o przyczynie katastrofy. **STALLED** oznacza, że leciałeś ze zbyt małą prędkością i spadłeś na ziemię, **CRASH LANDING** — że uderzyłeś w ziemię ze zbyt dużą siłą lub zapomniałeś wysunąć podwozia **MISSED RUNWAY** oznacza, że nie trafiłeś w pas startowy bądź zjechałeś z niego. **GROUND LOOP** mówi, że zbyt gwałtownie chciałeś skręcić w czasie kołowania.

W SOLO FLIGHT każde lotnisko jest zdefiniowane jako przecięcie się dwóch namiarów z dwóch stacji VOR. Dzięki temu łatwo jest je znaleźć.

W stanie Kansas jest 7 lotnisk, których namiary wynoszą (VOR1-VOR2): Wichita 222-001, Lyons 252-336, Emporia 225-022, Chanute 154-052, Alina 295-335, Topeka 330-016, Kansas City 065-036. Kansas jest płaskim terenem wprost idealnym dla początkujących pilotów, ale uważaj na tajemnicze piramidy indiańskie.

Drugim stanem jest Washington. Przebiega przez niego pasmo gór! Niektóre szczyty wznoszą się na 4000 stóp. Namiary dla lotnisk tego stanu są następujące: Portland 223-001, Salem 224-278, Kelso 251-350, Olympia 284--344, Seattlte 314-010 wysokość 500 stóp, Chelan 060-035 wys. 1000 stóp i Yakima 142-059 na 500 stopach.

W Kolorado, w Górach Skalistych, lotniska są prowizoryczne i niemal prymitywne. Dane nawigacyjne dla tego stanu wynoszą: Aspen 223-001 2000 stóp, Pueblo 143-074 1000 stóp, Glenwood 264-343 2000 stóp, Vail 184-030 2500 stóp, Boulder 053-037 500 stóp i Steamboat 334--008 na 2000 stóp.

Na koniec parę porad dla pilota:

1. Podczas kołowania nie wykonuj ostrych skrętów gdyż możesz zniszczyć delikatne podwozie.
2. Przed startem dokładnie zaplanuj trasę lotu.
3. W czasie lotu staraj się nie puszczać silnika na pełną moc, gdyż może on ulec przegrzaniu. Pełnej mocy używaj tylko przy starcie.
4. Nie przeładuj samolotu (bierz około dwóch worków). Samolot przeładowany jest bezwładny, potrzebuje dłuższej drogi lądowania i startu. Będzie ciężko mu się oderwać od lotnisk Kolorado.

Życzę Ci przyjemnej zabawy oraz tyle lądowań ile startów!

Firma: Micro Prose
Komputer: Atari XL/XE, Commodore 64/128

(gustav)



Liczę na Waszą pomoc. Szukam opisu do gry BARBARIAN na Atari 65 XE. Nie znam kodu, który pozwala uruchomić tę grę. W zamian za informacje oferuję opisy do innych gier.
Tomasz Gałecki ul. Gościnna 4 32-510 Jaworzno

Mam Atari XE. Szukam opisów do gier: MONTEZUMAS REVENGE, ZORRO i CHIMERA. W ZORRO chodzi mi głównie o nieśmiertelność.
Marcin Rogulski ul. Kamienna 12 97-300 Piotrów Tryb.

Poszukuję opisów do gier: COMMANDO, GLADIATOR, HOBBIT, BARBARIAN na Atari.
Tomek Frąckowiak os. Widok 17/24 66-200 Świebodzin

Nie wiem, jak uzyskać nieśmiertelność w grze COMMANDO na Spectrum+ . Pomóżcie!
Marcin Kwapisz
ul. R. Barana 25 m 39 42-200 Częstochowa

Proszę o dokładny plan i opis do gry CAVERNS OF ERIBAN na Atari.
Arek Ocipiński ul. Górna 25 47-330 Zdzieszowice

Kto mi przyśle opis JAMESA BONDA na Spectrum? W zamian opisy do: COMMANDO, PIPELINE, PARADISE i innych.;
Maciej Sionkowski ul. I Armii W.P. 23/2 22-100 Chełm

Mam Spectrum-K Proszę o pomoc w grach: ALCHEMIST, ZZOOM, SABRE WOLF, CYCLON (jak znaleźć wszystkie paczki?).
Marcin Rozner ul. Koński Jar 8 m 90 02-785 Warszawa

Może ktoś z czytelników będzie w stanie mi pomóc — szukam instrukcji do gry SILENT SERVICE na Amstrada CPC464. W zamian służę nieśmiertelnościami do różnych gier na ten komputer.
Marcin Borowiecki os. Dolnośląska 100/62
97-400 Bełchatów

Kto mi napisze, jak w grze TRAP DOOR otworzyć sejf ze skarbami?
Tomasz Saja ul. Szymanowskiego 7/14
58-506 Jelenia Góra

Nie mogę dać sobie rady z dwoma grami: GYRUSS (proszę o nieśmiertelność) oraz PITFALL II (proszę o rysunek i opis).
Grzegorz Chyrc ul. Szujskiego 1/15 31-123 Kraków

Chciałbym nawiązać kontakt z posiadaczami gier: RAM- BO, GLADIATOR, BARBARIAN, ATIC ATAC na ZX Spectrum.
Piotr Jedziński ul. Fiszera 7/4 80-231 Gdańsk

Poszukuję sekretnego kodu do gry KENNEDY APPRO- ACH. W zamian służę opisem gry ZORRO i innymi. Komputer Atari.
Mariusz Gęborek 32-631 Dankowice 545 woj. bielskie

Od wielu miesięcy poszukuję gry GHOSTS'N GOBLINS bez rezultatu. Bez Waszej pomocy nie zdobędę jej. Pomocy! Mam Atari.
Grzegorz Woroniak ul. Kościuszki 7/11 12-200 Pisz

Kto udostępni mi ciekawe gry na Amstrada? Jestem inwalidą, więc sam nie mogę ich zdobyć. Mam też problem z grą BRUCE LEE — nie wiem, jak zabić przeciwnika w połowie planszy. Bardzo proszę o pomoc.
Michał Kardasiński ul. Klonowa 46/31 25-538 Kielce

Szukam nieśmiertelności do gry KOKOTONI WILF — w zamian oferuję opis do ROBIN HOODa.
Radek Rusiński ul. Gw. Ludowej 3 16-400 Suwałki

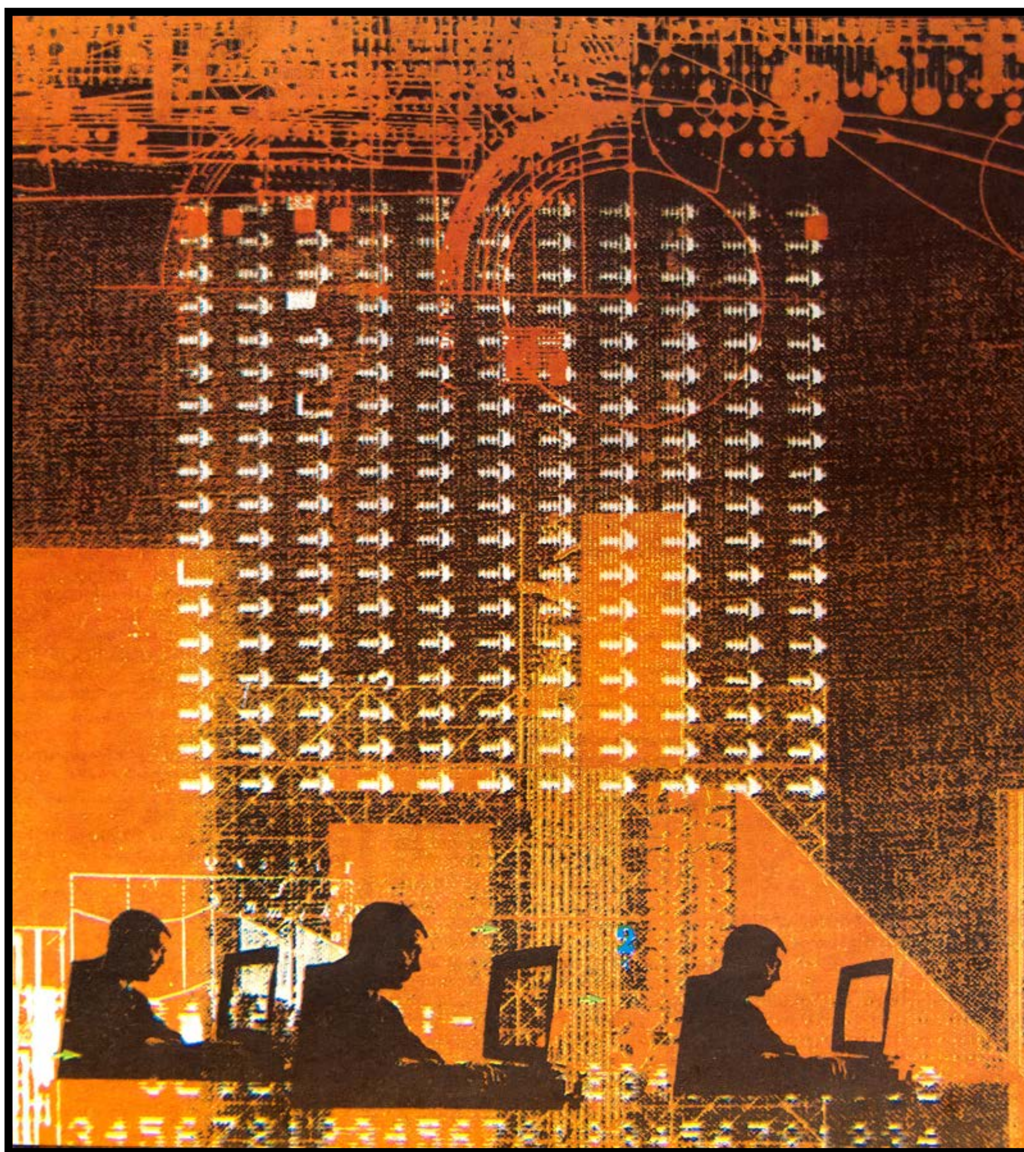
Jak pokonać smoka ziejącego ogniem w grze LAST NINJA i TRIAD w wersji dyskowej 175 linii na C64. Proszę też o krótki opis pokonywania trudniejszych miejsc w dalszych etapach tej gry, np. jak otworzyć bramę lub przejść przez skały.
Dariusz Pietrowiak ul. Newtona 3 B m 41
60-161 Poznań

Proszę, pomóżcie mi w grach: ZORRO, W.A.R., SPY Vs SPY, RAMBO. Wersja na C-64.
Zosia Rupala ul. Orkana 9 m 50 25-548 Kielce

Jak uzyskiwać nieśmiertelność w grach na ATARI 65 XE?
Paweł Lamik os. 2-go Pułku Lotniczego 19/147 31-868
Kraków-Nowa Huta

CZY KOMPUTER UMIE LICZYĆ?

(czyli zupa z gwoździa, którą każdy powinien zjeść, nawet jeśli mu nie smakuje)



To, że komputer potrafi liczyć bardzo szybko nie budzi niczyich wątpliwości. Spróbujmy się jednak zastanowić jak wygląda dokładność tych obliczeń.

Zaczniemy jednak od obliczeń wykonywanych ręcznie, na kartce kratkowanego papieru. Będziemy je wykonywać bardzo starannie — w jedną kratkę wolno wpisać tylko jedną cyfrę. Weźmy wąski pasek papieru, np. szerokości 5 kratek, liczby zapisujemy normalnie, w systemie dziesiętnym, jedynki pod jedynkami, dziesiątki pod dziesiątkami, itd., aby wygodnie było wykonywać działania (dokładnie tak uczono mnie rachunków w szkole podstawowej, więc nie będzie to nic trudnego). Dodajemy dwie liczby, np:

$$\begin{array}{r} 1234 \\ 128 \\ \hline 1362 \end{array}$$

Świetnie, ale tak naprawdę, to zrobiliśmy to działanie tylko aby upewnić się, że jeszcze potrafimy dodawać, gdyż zaraz zaczną się kłopoty. Dodajemy teraz naszym „przyrządem" 90123 i 82314:

$$\begin{array}{r} 90123 \\ 82314 \\ \hline 172437 \end{array}$$

Wynik, który otrzymaliśmy na pięciokratkowej kartce (72437) jest beznadziejnie zły. Oczywiście wiemy, że z jego lewej strony powinna stać jeszcze cyfra 1. Ale zapisać ją możemy co najwyżej na obrusie, z lewej strony kartki. Jeśli jednak nasz wynik zechcemy zabrać do innego pokoju, to straci wartość.

Co w takim razie zrobić? Rozwiązanie najprostsze: wziąć kartkę szeroką na sześć kratek. Świetnie, w takim razie ja poproszę, żebyście dodali 999999 i 901039. Widać, że ten sam problem się .powtórzy. Nie należy się poddawać zbyt łatwo. Bierzemy kartkę na 6 kratek, ale do wykonywania dodawań przyjmujemy tylko liczby pię- ciocyfrowe. Jeszcze lepiej, a ja w takim razie poproszę, żebyście liczbę pięciocyfrową, np. 90000 dodali do siebie 20 razy. Właściwie możecie sobie nawet oszczędzić trudu, bo 20 x 9000 = 1800000, czyli jest to liczba siedmiocyfrowa. Oczywiście można wziąć kartkę szeroką na 8 kratek, potem na 10, potem na 1000, ale przecież ta metoda do niczego nie prowadzi, zarówno pokój, w którym wykonujecie te arytmetyczne eksperymenty jak i kula ziemska, mają skończone rozmiary, natomiast liczbymogą rosnąć w nieskończoność. Zawsze więc znajdzie się liczba, dla której zabraknie kratki.

Jeśli ktoś ma ochotę powtórzyć tę samą zabawę w dwójkowym systemie liczenia, to proszę bardzo, gdyż zbliża nas to do celu zasadniczego — arytmetyki komputera. Dodawanie dwójkowe jest bardzo proste, popatrzmy na przykład (jeśli ktoś bardzo nie chce dodawać liczb binarnych, to może pominąć parę najbliższych akapitów):

$$\begin{array}{r} 1010 \\ 1101 \\ \hline 10111 \end{array}$$

a teraz na reguły:

$$\begin{array}{cccc} 0 & 0 & 0 & 1 \\ +0 & +1 & +0 & +1 \\ \hline 0 & 1 & 1 & 0 \end{array}$$

dodajemy tak samo jak w systemie dziesiętnym: podpisujemy liczby pod sobą i zaczynamy dodawanie od strony prawej (cyfry najmniej znaczące) do lewej. Bierzemy parę cyfr i wyznaczamy ich sumę. Możliwe są tylko cztery kombinacje cyfr, wynik dla każdej z nich napisałem kilka wierszy wyżej. Zwróćmy uwagą, że tak jak w systemie dziesiętnym, może wystąpić przeniesienie (1+1 = 10) i otrzymaną w wyniku tego przeniesienia jedynkę trzeba dodawać do następnej z lewej kolumny. Jeszcze jeden przykład:

$$\begin{array}{r} 1011101 \\ +\ 101100 \\ \hline 10001001 \end{array}$$

(jeśli ktoś ma ochotę sprawdzić, to można przeliczyć argumenty i wynik z systemu dwójkowego na dziesiętny — pisaliśmy już kilkakrotnie jak to zrobić).

Najważniejsze jednak jest to że znowu pojawiło się nam przeniesienie. Powstaje nowa cyfra, która nie mieści się w kratkach. Problem, który mieliśmy z za wąską kartką będzie również problemem komputera. Dużo bardziej dotkliwym, gdyż my moglibyśmy próbować brać większą kartkę, a komputer nie. Ma tyle miejsca (tyle bitów, czyli cyfr binarnych) na zapisanie liczby, ile przewidział konstruktor i zrobiła fabryka. Cóż może więc zrobić komputer gdy w wyniku działania arytmetycznego powstanie liczba, która nie da się zapisać w komórce pamięci (bo ma więcej cyfr niż jest bitów w komórce)? Może tylko zasygnalizować ten smutny fakt użytkownikowi i zakończyć obliczania przed czasem Tak właśnie zdarzyło się nam w zeszłym miesiącu podczas obliczania wartości wielomianu. Zwróćmy uwagę, że w wielomianie występuje wyraz anx". Gdy podaliśmy an = 100, x=5, n = 100, wyraz ten wynosił: 100 5100. Zgodzicie się chyba, że taka liczba miała prawo nie zmieścić się w zakresie dopuszczalnych wielkości, i stąd zamiast wyniku otrzymaliśmy sygnalizację: „Runtime error 01".

A może warto zastanowić się jakiej wielkości liczby można zapisać w pamięci komputera? Proponuję znowu zacząć od kartki w kratkę i liczb dziesiętnych. Weźmy jako przykład liczbę czterocyfrową. Oczywiście największą liczbą dziesiętną, którą można zapisać na czterech cyfrach jest 9999. Napiszmy ją w kratkach oznaczonych kolejnymi potęgami dziesiątki, a następnie dodajmy do niej 1:

$$\begin{array}{ccccc} 10^4 & 10^3 & 10^2 & 10^1 & 10^0 \\ & 9 & 9 & 9 & 9 \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}$$

Zrobiliśmy, niezbyt oryginalne, odkrycie, że 9999 + 1 = 10000. Ale nie tylko. Możemy bowiem łatwo zauważyć, że liczba mająca 4 cyfry, zwiększona o 1 daje 10^4. Czyli inaczej 99 99 = 10^4 - 1. Dokładnie tak samo możemy rozumować dla liczb mających 5 cyfr (największa z nich, to 9999 = 10^5 - 1), i w ogóle dla liczb dowolnej długości. Krótko mówiąc, nasze odkrycie jest następujące: jeśli największą liczbę mającą n cyfr oznaczyć przez X_n, to $X_n = 10^n - 1$.

Czy warto zawracać sobie głowę tym, że 99 = 10^2 - 1, czy też problemem jaka jest wartość największej oś- miocyfrowej liczby? Rozstrzygnęliśmy ten problem w systemie dziesiętnym, gdyż jest to system dużo przystępniejszy dla przeciętnego człowieka niż binarny. Dzięki temu łatwiej nam będzie zrozumieć jaka największa liczba da się zmieścić na n bitach. Będzie to oczywiście liczba składająca się z samych jedynek (tak jak w dziesiętnym

z dziewiątek), czyli 1111, dodajmy do niej 1:

$$\begin{array}{ccccc} 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}$$

Widzimy dokładnie to samo zjawisko: 1111 = 2^4 - 1, i dalej, jeśli największą liczbę dwójkową o n cyfrach (dwójkowych) oznaczymy przez B_n, to mamy $B_n 62_n$ - 1.

Teraz tylko trzeba raz jeszcze przypomnieć, że jednej cyfrze dwójkowej odpowiada jeden bit w pamięci komputera, (czyli np. w ośmiobitowym bajcie zapisujemy ośmiocyfrową liczbę) i widać dokąd zmierzaliśmy. Jeśli wiemy ile bitów przeznaczono w komputerze na zapisanie każdej liczby łatwo obliczyć jaka największa wartość może być zapisana. Np. dla wspomnianego bajtu będzie to wartość 2^8 - 1 = 255. Gdyby na każdą liczbę przeznaczyć nie jeden a dwa bajty (8 2 = 16 bitów) największą liczbą będzie 2^{16} - 1 = 65535, gdyby przeznaczyć 4 bajty (32 bity), to można reprezentować liczby od 0 do 4 294 967 296. Warto zwrócić uwagę na kilka rzeczy: po pierwsze, ta ostatnia liczba, choć na pozór ogromna, wcale nie zawsze musi wystarczyć, jeśli np. zechcemy zsumować nogi wszystkich żyjących na świecie ludzi, to okazuje się ona za mała na wynik. Po drugie, im więcej bajtów przeznaczymy na zapisanie w pamięci operacyjnej komputera jednej liczby, tym mniej liczb można tam wpisać, więc dodawanie bajtów w nieskończoność nie wchodzi w grę jako metoda zapisu bardzo dużych liczb. Po trzecie wreszcie, żeby nasz komputer nie zgłupiał przy rachunkach, dobrze było by, żeby wszystkie liczby były zapisywane na takiej samej liczbie bajtów. Wadą tego rozwiązania jest z kolei marnotrawstwo miejsca gdy obok liczb dużych (wymagających np. 8 bajtów) występują w obliczeniach niewielkie, mieszczące się na jednym czy dwóch.

Jednym słowem same kłopoty. A na domiar złego dalej nie wiemy dlaczego miesiąc temu obliczając wartości wielomianu dostaliśmy dla tych samych danych kilka różnych wyników. Wyraźnie więc, w naszym opisie arytmetyki komputera czegoś jeszcze brakuje.

Tak naprawdę, to brakuje nawet paru rzeczy. Jedną z nich są liczby ujemne. Jak je wprowadzić? Najprostszą metodą jest poświęcić w komórce pamięci w której zapisujemy liczbę jeden bit na znak. Jest to rozwiązanie powszechnie stosowane, a jego łatwą do odgadnięcia konsekwencją jest zmniejszenie zakresu liczb, które możemy zapisać. Jeśli na liczbę mamy 16 bitów i jeden przeznaczymy na znak, to na zapis cyfr pozostanie nam 15. Jaką największą liczbę można zapisać na piętnastu cyfrach. Wszyscy już doskonale wiedzą że 2^{15} - 1 = = 32767. Przy czym jeżeli znak jest „ -”, to będzie to -32767, jeżeli „ +”, to 32767. Czyli ostatecznie możemy przetwarzać liczby całkowite z przedziału od -32767 do 532767.

Taka metoda zapisywania liczb ze znakiem jest bardzo naturalna, i dlatego właśnie ja wybrałem ją z pośród kilku możliwych rozwiązań tego problemu. Pozostałe są trochę trudniejsze do zrozumienia i darujemy je sobie aby nie utknąć w gąszczu szczegółów technicznych.

Jeśli zaś chodzi o pozostałe braki w naszej wiedzy o tym jak komputer liczy, to postaramy się uzupełnić je za miesiąc.

Andrzej Pilaszek

dokończenie ze str. 15

```
10 'Notatnik                            [884]
20 CLS:MODE 2:DIM                       [1978]
a$(100),b$(100),c$(100)
30 CLS:LOCATE 30,3:PRINT "M E N         [3837]
U":LOCATE 30,4:PRINT "-------"
40 LOCATE 20,5:PRINT "1 - Zakladanie    [2471]
notatnika"
50 LOCATE 20,7:PRINT "2 - Szukanie      [2603]
informacji"
60 LOCATE 20,9:PRINT "3 - Zapis na      [4384]
dyskietke"
70 LOCATE 20,11:PRINT "4 - Odczyt       [4059]
z dyskietki"
80 LOCATE 20,13:PRINT "5 -              [1736]
Modyfikacja notatek"
90 LOCATE 20,15:PRINT "6 -              [2689]
Kasowanie informacji"
100 LOCATE 10,22:INPUT "Podaj           [3537]
odpowiednia cyfre : ",k
110 ON k GOTO                           [2224]
130,240,400,470,600,810
120 LOCATE 10,22:PRINT STRING$(50,"     [2243]
"):GOTO 100
130 CLS:FOR i=1 TO 100                  [1259]
140 CLS:LOCATE 20,3:PRINT               [2568]
"ZAKLADANIE NOTATNIKA"
150 IF a$(i)="" THEN 160 ELSE 230       [1259]
160 LOCATE 10,5:INPUT "Wprowadz         [4482]
nazwisko i imie : ",a$(i)
170 LOCATE 10,7:INPUT "Wprowadz         [2446]
adres : ",b$(i)
180 LOCATE 10,9:INPUT "Wprowadz         [3850]
numer telefonu : ",c$(i)
190 LOCATE 10,22:PRINT "Nastepne        [4392]
nazwisko (T/N) ? : "
200 q$=INKEY$:IF q$="" THEN 200         [3061]
ELSE q$=UPPER$(q$)
210 IF q$="T" THEN 230                  [1206]
220 IF q$="N" THEN 30 ELSE 200          [1579]
230 NEXT                                [350]
240 CLS:GOTO 540                        [1021]
250 CLS:LOCATE 20,3:PRINT "Szukanie     [2449]
informacji"
260 LOCATE 10,10: INPUT "Wprowadz       [3511]
nazwisko i imie : ",a$
270 FOR i=1 TO 100                      [326]
280 IF a$=a$(i) THEN 330                [327]
290 NEXT                                [350]
300 LOCATE 15,22:PRINT "W notatniku     [4490]
nie ma takiego nazwiska"
310 FOR a=1 TO 1000:NEXT                [1114]
320 GOTO 30                             [340]
330 CLS:LOCATE 16,5:PRINT "Nazwisko     [4516]
i imie : ";a$(i)
340 LOCATE 10,7:PRINT "Adres :          [3157]
";b$(i)
350 LOCATE 10,9:PRINT "Numer            [2730]
telefonu : ";c$(i)
360 LOCATE 10,22:PRINT "Szukasz         [3956]
dalej (T/N) ? : "
370 q$=INKEY$:IF q$="" THEN 370         [1812]
ELSE q$=UPPER$(q$)
380 IF q$="T" THEN 240                  [1173]
390 IF q$="N" THEN 30 ELSE 370          [1623]
400 CLS:LOCATE 20,15:PRINT "Uwaga!      [5898]
Trwa zapis na dyskietke."
410 LOCATE 20,3: PRINT "Z A P I S"      [2126]
420 OPENOUT "notki"
430 FOR i=1 TO 100                      [1688]
440 WRITE #9,a$(i),b$(i),c$(i)          [326]
450 NEXT                                [2339]
460 CLOSEOUT:GOTO 30                    [350]
470 CLS:LOCATE 26,15:PRINT "Uwaga!      [1533]
Trwa odczyt z dyskietki."               [3445]
480 LOCATE 20,3:PRINT "O D C Z Y        [2192]
T "
490 OPENIN "notki"                      [1229]
500 FOR i=1 TO 100                      [326]
510 INPUT #9, a$(i),b$(i),c$(i)         [1110]
520 NEXT                                [350]
530 CLOSEIN:|ERA,"notki":GOTO 30        [2139]
540 LOCATE 20,3:PRINT "W Y S Z U K      [2727]
I W A N I E"
550 LOCATE 20,5:PRINT "1 - Nazwiska"    [2132]
560 LOCATE 20,7:PRINT "2 - Lista"       [1643]
570 LOCATE 10,22:INPUT "Podaj           [3537]
odpowiednia cyfre : ",k
580 ON k GOTO 250,950                   [1387]
590 LOCATE 10,22:PRINT STRING$(50,"     [2269]
"):GOTO 570
600 CLS:LOCATE 20,3:PRINT "M O D        [3390]
Y F I K A C J A   I N F O R N A C
J I"
610 LOCATE 10,10:INPUT "Poszukiwane     [4097]
nazwisko i imie : ",x$
620 FOR i=1 TO 100                      [326]
630 IF x$=a$(i) THEN 670                [1623]
640 NEXT                                [350]
650 LOCATE 10,22:PRINT "W notatniku     [4773]
nie ma takiego nazwiska !"
660 FOR b=1 TO 1000:NEXT:GOTO 30        [2926]
670 CLS:LOCATE 20,3:PRINT "M O D Y      [1754]
F I K A C J A"
680 LOCATE 20,5:PRINT "Nazwisko :       [1969]
";a$(i)
690 LOCATE 10,7:INPUT "Wpisz            [3201]
poprawione nazwisko i ";a1$
700 IF a1$<>"" THEN a$(i)=a1$           [2010]
710 LOCATE 20,9:PRINT "Adres :          [2378]
";b$(i)
720 LOCATE 10,11:INPUT "Wpisz           [3175]
aktualny adres : ";b1$
730 IF b1$<>"" THEN b$(i)=b1$           [1072]
740 LOCATE 20,13:PRINT "Numer           [2495]
telefonu : ";c$(i)
750 LOCATE 10,15:INPUT "Wpisz           [4445]
aktualny numer telefonu : ";c1$
760 IF c1$<>"" THEN c$(i)=c1$           [1027]
770 LOCATE 10,22:PRINT "Czy             [3750]
modyfikujesz dalej (T/N) ? :"
780 q$=INKEY$:IF q$="" THEN 780         [2709]
ELSE q$=UPPER$(q$)
790 IF q$="T" THEN 600                  [1197]
800 IF q$="N" THEN 30 ELSE 780          [1559]
810 CLS:LOCATE 26,3:PRINT "K A S        [4478]
O W A N I E   I N F O R M A C J I"
820 LOCATE 20,5:INPUT "Poszukiwane      [3853]
nazwisko i imie : ";xx$
830 CLS:FOR i=1 TO 100                  [1259]
840 LOCATE 20,3:PRINT "K A S            [3789]
O W A N I E   I N F O R M A C J I"
850 IF xx$=a$(i) THEN 890               [2056]
860 NEXT                                [350]
870 LOCATE 10,22:PRINT "W notatniku     [4773]
nie ma takiego nazwiska !"
880 FOR c=1 TO 1000:NEXT:GOTO 30        [2113]
890 LOCATE 20,5:PRINT "Nazwisko :       [1969]
";a$(i)
900 LOCATE 20,7:PRINT "Czy chcesz       [3752]
wykreslic to nazwisko (T/N) ? : "
910 q$=INKEY$:IF q$="" THEN 910         [2243]
ELSE q$=UPPER$(q$)
920 IF q$="T" THEN 940                  [1096]
930 IF q$="N" THEN 30 ELSE 910          [1722]
940 a$(i)="":b$(i)="":c$(i)="":GOTO     [2241]
30
950 CLS:k=0:LOCATE 30,1:PRINT "L        [1435]
I S T A"
960 LOCATE 3,1:PRINT "Nazwisko          [6229]
i imie":LOCATE 50,1:PRINT
"Adres":PRINT STRING$(80,"_")
970 FOR i=1 TO 100                      [326]
980 IF a$(i)<>"" THEN GOSUB 1010        [2196]
990 NEXT                                [350]
1000 PRINT                              [1695]
STRING$(80,CHR$(208)):GOTO 1060
1010 k=k+1:LOCATE 1,1+k*2:PRINT         [2590]
CHR$(211)
1020 LOCATE 3,1+k*2:PRINT               [3723]
a$(i):LOCATE 47,1+k*2:PRINT
CHR$(211)
1030 LOCATE 50,1+k*2:PRINT              [2765]
b$(i):LOCATE 80,1+k*2:PRINT
CHR$(209)
1040 IF 1+k*2=21 THEN 1000              [594]
1050 RETURN                             [555]
1060 LOCATE 40,24:PRINT                 [4942]
CHR$(24);"Wcisnij dowolny
klawisz":CHR$(24)
1070 CALL &BB18:CLS:GOTO 30             [1616]
1080 LOCATE 1,22:PRINT                  [3045]
STRING$(80,CHR$(208))
1090 LOCATE 40,24:PRINT                 [5105]
CHR$(24);"Wcisnij dowolny
klawisz";CHR$(24)
1100 CALL &BB18:IF i=100 THEN 30        [2906]
ELSE CLS:k=0
1110 LOCATE 30,1:PRINT "L I S T         [3392]
A":PRINT STRING$(80,"_"):GOTO 1090
```

CeBIT '88

DRUGIE SPOJRZENIE

Moje wrażenia z CeBIT są trochę inne od tych, które miałem okazję przeczytać dotychczas, choćby w majowym „Bajtku". Myślę, że ogrom targów wyjaśnia te niezgodności. Przy tak wielkiej imprezie nie tylko dwie, ale chyba i dziesięć osób może opowiadać coś zupełnie innego i nie jest to sprzeczność. Ja, wybaczcie, chciałbym opowiedzieć Wam tym razem nie o nowościach Commodore, czy Atari.

Co to jest CeBIT? Odbywające się co roku w Hanowerze (RFN) wielkie targi. Czy komputerowe? Popatrzmy na jakie grupy organizatorzy podzielili ekspozycję:

1. Office and Information systems.
2. Banking and Security systems.
3. Software and Consulting.
4. DP Perpherals.
5. Office and Organizational Technology.
6. Applications Center for Small to Midsized Companies and Professionals.
7. Telecommunication, Research and Development.
8. C-Technologies, Production Data Aquisition.*)

Podaję w wersji oryginalnej zachowując nawet szczegóły pisowni, aby uniknąć posądzenia o tendencyjne tłumaczenie. A trzeba mieć argument, gdy chce się powiedzieć coś sprzecznego z obiegowym poglądem. **CeBIT to nie są targi komputerów.** Organizatorzy używają nazwy: „The World Center for Office, Information and Telecommunication". **Oferowane na nich jest wszystko, co może służyć sprawnemu przetwarzaniu informacji, ułatwieniu pracy w biurze, sklepie czy banku, organizacji łączności. Rzeczywiście; bardzo wielu wystawców przywiozło ze sobą komputery. Często nie są one jednak towarem samym w sobie, lecz elementem systemu przeznaczonego do wykonywania jakiegoś zadania, np. prowadzenia korespondencji, gromadzenia danych.**

Podkreślam, elementem. Typowy wystawca nie oferuje sprzętu na zasadzie: kup Pan komputer bo to bardzo pożyteczna maszyna, na pewno Pan wymyśli dla niego jakieś zastosowanie. Oferowane są zestawy realizujące jakąś funkcję, rozwiązanie konkretnego problemu. Np.: „nasz system zastępuje karty zegarowe przy rejestracji czasu pracy, automatycznie rozlicza pracowników, itd...", a nie: „nasz system zbudowany jest w oparciu o 32-bitowy mikrokomputer". Rozsądny dyrektor, który ma problem rozliczania czasu pracy dla trzech tysięcy pracowników, kupi system działający nawet na komputerze 14-i-pół-bi- towym, pod warunkiem, że będzie dobrze robił to co powinien. Natomiast nawet najwspanialszy komputer, tysiącbitowy, pełen megabajtów, megahercy i do tego jeszcze turbo do kwadratu, jest dla niego zupełnie nic nie wart jako urządzenie samo w sobie.

Komputery też były

Żeby uniknąć nieporozumień dodaję, że oczywiście, wystawiają się także producenci sprzętu. I ci wielcy, znani w całym świecie, i wielu takich, których pierwszy raz widziałem na oczy. Jednak szukają oni przede wszystkim klientów innego rodzaju. Choćby właśnie wspomniani przed chwilą producenci systemów użytkowych muszą gdzieś kupować komputery do tych systemów. Hurtownicy, właściciele sklepów to są partnerzy dla poważnych producentów. Oczywiście wielkie stoiska rekinów urządzane są także z myślą o zwykłych użytkownikach, którzy chcą zobaczyć nowości mające pojawić się w handlu w najbliższej przyszłości. Wszyscy goście są witani uprzejmie — reklama jest dźwignią handlu. Każdy chce zrobić najlepsze wrażenie, jeśli jego komputery spodobają mi się, to właśnie o nie będę później pytał swoich lokalnych dostawców sprzętu.

Bardzo wielu dostawców techniki biurowej, handlowej czy jeszcze innej oferowało rozwiązania bazujące na ogromnych możliwościach współczesnych komputerów, a przede wszystkim mikrokomputerów. Stąd wrażenie dominacji sprzętu liczącego w wielu stoiskach. Jednak **wystawiane jest również wiele urządzeń zupełnie nic nie mających wspólnego z komputerem. Podręczne maszynki do szybkiego oprawiania dokumentów z pełną gamą kolorowych okładek do wyboru, a z drugiej strony niszczarki, zamieniające wkładane do nich papiery na drobniutką sieczkę. Wielkie szafy z obracanymi silnikami pojemnikami na tradycyjne papierowe dokumenty, i malutkie dyktafony. Całe systemy przygotowywania i wykorzystywania mikrofilmów oraz maszyny do otwierania kopert. A także cała gama zwykłych (czyli elektrycznych!) maszyn do pisania, także takich dość inteligentnych, oraz spora oferta elektronicznych kalkulatorów.**

Nie tylko biuro

Także inne dziedziny działalności człowieka miały swoje tematyczne ekspozycje. Oczywiście znalazła odbicie ogromna rola pieniądza w naszym życiu. Sprzęt dla banków miał odrębną ekspozycję. Oprócz dużych skomputeryzowanych systemów obsługujących operacje banku były też i maszyny liczące bilon, czy wymieniające pieniądze (np. funty, franki i dolary na marki). Specjalizowane terminale kasowe i rachunkowe, systemy elektronicznych zabezpieczeń.

Pawilony łączności zawierały pełną gamę propozycji: od malutkich przenośnych radiotelefonów, do anten satelitarnych i systemów zapewniających łączność na całej kuli ziemskiej. Dla poważnych ludzi interesu oferowano wbudowane w biurko systemy wielokanało-wej łączności z całym światem, sprzężone z komputerem i Bóg raczy wiedzieć czym jeszcze. **Nie tylko dla nich są natomiast telefaksy — urządzenia do przesyłania na odległość kopii dokumentów. Wykorzystują zwykłą linię telefoniczną i są chyba bardzo użyteczne, gdyż wszystkie firmy z którymi miałem do czynienia oprócz telefonu i teleksu podają numer telefaksu.**

Także handlowcy, hotelarze i właściciele restauracji nie mogli narzekać na brak zainteresowania swoimi potrzebami. Elektroniczne kasy rejestracyjne, samodzielne i współpracujące z komputerami, wagi elektroniczne, kompletne stanowiska kasowe dla domów towarowych. Wszystko to oferowano na przynajmniej kilkudziesięciu stoiskach.

I wreszcie sprzęt chyba głównie biurowy, ale zasługujący na oddzielną wzmiankę: kopiarki. Najwięksi, np. Canon czy Minolta wystawili ich po kilkanaście typów. Duże i małe, potrafią robić powiększenia i zmniejszenia, pracować w kilku kolorach, sortować kopie jeśli robimy kilka egzemplarzy. Są nawet egzemplarze przenośne, na baterie.

Do długotrwałej, właściwe} pracy komputer potrzebuje nie tylko odpowiedniej jakości zasilania z sieci elektrycznej, ale i wielu dodatkowych materiałów. Dyskietek, taśmy barwiącej do drukarki, pisaków do plotera. Różnych rodzajów papieru do drukarki, pojedynczego lub wielowarstwowego, najlepiej z kalką która nie brudzi wszystkiego dookoła na czarno. Wreszcie potrzebne są materiały do konserwacji: czyszczenia głowic, klawiatury, antystatyczne ściereczki i aerozole. Przy braku tego wszystkiego, w zaopatrzeniowej próżni komputery długo nie pociągną. Na całym świecie jest to oczywiste i zrozumiałe, prawda?

Kilku wystawców oferowało też siatki ochronne, zakładane na ekran. Mają one chronić organizm operatora (przede wszystkim wzrok) przed szkodliwymi skutkami pracy przy monitorze. Są one mało znane w Polsce, a być może warto się nimi zainteresować.

Szklana ściana

Na targach zainstalowany był komputerowy system informacyjny. Korzystać można było z niego bezpłatnie, ale za pośrednictwem operatorów, obsługujących terminale. Ci właśnie operatorzy dostarczyli

mi (zupełnie niechcący) nieoficjalnej listy najczęściej szukanych firm. Żeby ułatwić sobie pra cę przykleili na szybach przed sobą adresy firm, o które najczęściej pytano Oto lista, którą dla Was przepisałem: C**ommodore, Apple, Atari, Schneider, Mi crosoft, Data Becker, Markt und Technik, Hewlett Packard, NEC, Toshiba, Casio, Sharp, Epson, Zenith.** Jeśli ktoś nie wie kto to jest Markt und Technik proponuję, żeby się nie martwił, ja też nie wiem.

Część firm starafa się zwracać uwagę publiczności nie tylko swoją podstawową ofertą. Pisaliśmy już o pokazach ruchomych, trójwymiarowych obrazów. Zdarzały się też rzeczy mniej techniczne, np. występy mistrza magii lub koncert egzotycznych instrumentów przed stoiskiem firmy. Koncert składał się między innymi z pięknej gry na pile, ale nie był chyba tak atrakcyjny jak urodzinowe przyjęcie (piętnaste urodziny) firmy „Obra" robiącej oprogramowanie, gdyż tam, oprócz zespołu muzycznego występowała jeszcze wielka beczka piwa i trzej barmani wręczający gościom pełne kufle.

Na koniec chciałbym jeszcze wrócić do rzeczy poważniejszych. Targi były niemieckojęzyczne, jednak sporo materiałów można było uzyskać po angielsku. Prawie wszystkie osoby, do których zwracałem się po angielsku, płynnie odpowiadały w tym języku. Prawdziwym powodem, dla którego podałem „spis treści" wystawy w oryginale jest chęć zwrócenia Waszej uwagi na ten problem. Osoby nie znające żadnego języka poza polskim mogą sobie próbować wyobrazić targi tak: kilkanaście tysięcy komputerów i innych urządzeń, przy nich jakieś napisy, stoją uśmiechnięci ludzie, którzy chętnie wszystko wyjaśniają. A my stoimy wśród nich, ale to nieprawda, że jesteśmy partnerami. Nie mamy żadnej możliwości porozumienia się, tak, jakby dzieliła nas gruba szyba. Nie piszę tego, aby komuś dokuczyć, ale raczej, aby zmobilizować wszystkich do nauki języków. Tak przy okazji, znajomość angielskiego rozwiązuje wiele problemów, np. nie trzeba czekać, aż ktoś przetłumaczy interesującą nas książkę. No i oczywiście, za'kilka lat pojedziemy razem na CeBIT. Zapewniam Was, że warto.

Jonasz Mayer

*) C — technologie, to wszystkie nowe techniki pracy mające w nazwie „computer": CAD — Computer Aided Design, CAE — Computer Aided Engineering, CIM — Computer Integrated Manufacturing, czyli, ogólnie mówiąc, wspomagane komputerem projektowanie i produkcja.





DLA POCZĄTKUJĄCYCH

Wydawnictwa Szkolne i Pedagogiczne wydały ostatnio dwie książki dla początkujących. „**Pierwsze kontakty z mikrokomputerem ZX Spectrum**" mają ułatwić opanowanie tajników obsługi Spectrum tym wszystkim, którzy nigdy nie mieli z nim do czynienia. Autorzy nie zamierzali umieścić tu wykładu ani opisu żadnego języka programowania. Zajęli się wyłącznie stroną użytkową i przedstawili informacje najniezbędniejsze do korzystania ze Spectrum. Zasób przekazywanych w książce wiadomości wystarcza do opanowania zasad obsługi klawiatury, wczytania i nagrania programów z magnetofonu. Kilka listingów prostych gier może obudzić w czytelniku zainteresowanie samodzielnym programowaniem i skłonić do sięgnięcia po inne publikacje z tej dziedziny.

"**Pierwsze kroki w LOGO**" jest książką, z którą warto zapoznać się zanim nauczymy się programować w Basicu. O wiele łatwiej będzie wtedy zrozumieć zasady niezwykle interesującego języka. Autor opisał wszystkie instrukcje występujące w LOGO, procedury, sposób używania zmiennych. Przykłady i zadania do samodzielnego rozwiązania mają rozbudzić inwencję twórczą czytelnika. Po przeczytaniu książki można z powodzeniem pokusić się o rozwiązanie nawet dość trudnych problemów. Jedyny warunek to chęć do pracy, pomysłowość i odrobina cierpliwości.

Obie pozycje kosztują w sumie zaledwie 320 zł. Nie jest to wiele, ale... Pierwsza z nich zawiera zdecydowanie za mało informacji. Staje się zbyteczna, gdy mamy dobrze przetłumaczoną instrukcję lub bardziej doświadczonego kolegę, który praktycznie pokaże w ciągu 15 minut, jak obsługiwać Spectrum.

(j.j.)

Emanuel Czyżo, Wacław Zawadowski „Pierwsze kontakty z mikrokomputerem ZX Spectrum", Wydawnictwa Szkolne i Pedagogiczne, Warszawa 1987, Wydanie I, Nakład 50.000 egz., cena 200 zł.

Jarosław Kania „Pierwsze kroki w LOGO", Wydawnictwa Szkolne i Pedagogiczne, Warszawa 1987, Wydanie I. Nakład 60.000 egz., cena 120 zł.

— KONKURS —

MATEMATYCZNY!!!

ZADANIE NR 1

Trójkątem pitagorejskim nazywamy trójkąt prostokątny o długości wszystkich boków wyrażonej liczbą całkowitą np 3:4:5 lub 5:12:13 itd.

W matematyce istnieje cały dział matematyki oparty na tym pomyśle. Dzisiaj proponuję ułożenie programu, który w ciągu dwóch minut znajdzie większą liczbę takich trójkątów. Oczywiście NIE należy brać pod uwagę trójkątów o bezpośrednim stałym mnożniku, których boki są proporcjonalne do boków mniejszego trójkąta np. 6:8:10 lub 15:36:39

Do rozwiązania należy dołączyć program wraz z opisem oraz podać liczbę i otrzymane trójkąty pita- gorejskie.

Program musi być napisany w języku BASIC. Dopuszczone są następujące instrukcje i funkcje: PRINT, GOTO, GOSUB, RETURN, IF...THEN, LET, DIM, FOR...MEXT...STEP, AND, OR, NOT, RND, ABS, INT. SQR, EXP, ATN, COS, SIN, TAN, LOG (logarytm naturalny). PI (liczba pi), SGN, +, -,*,/, t, >, >=, <, < = , =, o. UWAGA! Przy podstawianiu należy posługiwać się rozbudowaną instrukcją LET.
Ponadto od programu wymaga się:
1) program ma zaczynać się od linii 1 z przyrostem linii co 1
2) program zawiera po jednej instrukcji w linii.
3) obliczone boki trójkątów winny być przechowane w jawnie określonej tablicy
4) czas dwóch minut mierzy się od momentu wykonania pierwszego obliczenia (nie jest brany pod uwagę czas potrzebny na zadeklarowanie tablicy dlatego Redakcja prosi o umieszczanie takich deklaracji na początku programu i zaznaczenie początku obliczeń przez np. PRINT "START -->"

Jedyną instrukcją dopuszczoną więc przed instrukcją PRINT "START -->" jest DIM.

UWAGA!!! Wszystkie programy pretendujące do wygranej będą wpisane na jeden komputer by jak zwykle zwyciężyła metoda rozwiązania problemu a nie szybszy komputer. Zachęcamy więc do brania udziału w konkursie i przysyłania swoich rozwiązań. Ciąg dalszy konkursu już za miesiąc.

ZADANIE NR 2

Pytanie będzie bardzo krótkie choć pewnie ono przysporzy najwięcej kłopotów. Pytanie brzmi następująco:

Z jaką dokładnością została policzona liczba Eulera, jeśli suma wszystkich cyfr po przecinku dała w rezultacie 4091 (cztery tysiące dziewięćdziesiąt jeden)???

Liczba Eulera definiowana jest jako granica ciągu (1+(1/n))fn przy n dążącym do nieskończoności. Z dokładnością do 6 miejsc po przecinku liczba ta wynosi: 2.718281...... i jest używana między innymi jako podstawa logarytmów w komputerach. Tym, którzy nie pamiętają wyżej poruszonych tematów (granica) proponuję sięgnąć do podręczników matematyki z ostatnich klas szkoły podstawowej i z pierwszych dla szkół średnich.

Do rozwiązania należy dołączyć program (obliczający liczbę E z dokładnością do n miejsc po przecinku) wraz z opisem oraz podać dokładność obliczenia liczby Eulera.

Program musi być napisany w języku BASIC.

ALGORYTMY GENERACJI KRZYWYCH NA PŁASZCZYŹNIE

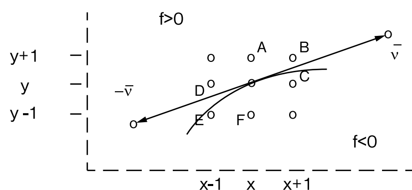
Wiele osób patrząc na figury geometryczne wyświetlane na monitorze typowego komputera zauważy, że okręgi nie są wcale okręgami, a grubość linii prostej zależy od jej nachylenia. Dyskretna struktura obrazu nakłada poważne ograniczenia na wygląd tych figur i zmusza do starannego doboru algorytmów ich kreślenia.

Większość algorytmów dotychczas proponowanych na lamach popularnych czasopism zajmujących się informatyką prezentowała rozwiązania najprostsze i stosunkowo mato efektywne. Przykładowo, sposób kreślenia okręgu oparty o wyliczanie dla każdego X współrzędnej Y wg wzoru: y = V r2 f' jest wyjątkowo kosztowny jeśli chodzi o czas obliczeń. Nawet w tak trywialnym przypadku, jak rysowanie linii prostej, pokutują algorytmy działające niedokładnie dla szczególnych nachyleń: 0° i 180°. Te właśnie metody zostały zastosowane w [1.2] i znane są w literaturze jako algorytmy Danielssona[3].

Bardzo efektywną klasę tworzą tzw. algorytmy przyrostowe [4]

Ich działanie można przedstawić na przykładzie prostego plotera. Przyjmijmy, że jedynymi rozkazami tego urządzenia są polecenia przesunięcia pióra w każdym z dwóch kierunków o ściśle określony krok: Zwiększ X o jeden, Zwiększ Y o jeden Zmniejsz X o jeden, Zmniejsz Y o jeden

Jeśli pióro plotera znajduje się w punkcie (x,y) (patrz rys. 1) to następne jego możliwe położenia mają współrzędne (x + Δx,y), (x,y+Δy) i (x+Δx,y+Δy), gdzie Δx,Δy=±1. Przy kreśleniu krzywej możemy przesuwać się wzdłuż jednego z dwóch kierunków (wektory v lub -v) Warunek ten pozwala ograniczyć liczbę wybieranych punktów do 3 (A,B,C lub D,E,F). Jeśli przyjmiemy dodatkowe uproszczenie o niemożliwości jednoczesnego przesunięcia pióra wzdłuż obu osi, to liczba wybieranych punktów zredukuję się do dwóch (A,C lub D.F). W tym momencie należy określić kryterium wyboru lepszego punktu, tzn. takiego, który leży bliżej rzeczywistej krzywej. W tym celu musimy przypomnieć sobie trochę matematyki.



Każdą krzywą na płaszczyźnie możemy opisać następującym równaniem: $f(x,y) = 0$

Dla okręgu o środku w punkcie (0,0) i promieniu R:

$f(x,y) = x^2+y^2-R^2 = 0.$

Oznaczmy przez: $f^x = f(x + \Delta x,y)$ i $f^y = f(x,y +\Delta y)$

wartości funkcji f odpowiednio w punktach przesuniętych o Δx i Δy.

Jeśli $| f^x | < | f^y |$, to przesuwamy się wzdłuż osi x, w przeciwnym wypadku wykonujemy przesunięcie wzdłuż osi y. Do wyznaczenia wzorów aktualizujących wartość funkcji f^x i f^y w zależności od wartości funkcji $f^a = f(x,y)$, posłużmy się rozwinięciem funkcji dwóch zmiennych w szereg:

gdzie:

są pierwszymi i drugimi pochodnymi funkcji f. Dla okręgu:

$f_x = 2x, f_y = 2y, f_{xx} = 2, f_{yy} = 2, f_{xy} = 0$

$f^x = f^a + 2*x*\Delta x+1,$

$f^y = f^a + 2*y*\Delta y+1.$

Jeśli będziemy poruszać się zgodnie z ruchem wskazówek zegara to:

Δx= +1, gdy fy>0

-1 gdy fy<0

Δy = +1, gdy fx<0

-1, gdy fx=>0.

Zauważmy, że wzory otrzymane dla okręgu mają wyjątkowo prostą postać. Najbardziej skomplikowane operacje to mnożenie przez dwa i dodawanie dwóch liczb i jedynki. W systemie dwójkowym mnożenie przez dwa odpowiada przesunięciu w lewo o jeden bit. Wszystkie te operacje odpowiadają podstawowym rozkazom mikroprocesorów. Mnożenia przez Δx lub Δy nie ma gdyż wielkości te są równe ±1.

Wykorzystanie powyższych wzorów razem z regułami wyboru Δx, Δy pozwala napisać następującą wersję procedury kreślenia okręgu o środku w punkcie (0,0) i promieniu R

```
procedure okrąg (r: integer):
var
  x,y, dx,dy,fa,fx,fy: integer;
begin
x : = r; y := 0, fa : = 0;
repeat
  Plot (x,y);
  if x>0 then begin dy := -1; fy := fa - y shl 1 + 1 end
        else begin dy := 1; fy := fa + y shl 1 + 1 end;
  if y>0 then begin dx := 1; fx := fa + X shl 1 + 1 end
        else begin dx := -1; fx := fa - x shl 1 + 1 end;
    if abs(fx)<abs(fy)
    then begin fa := fx; x := x + dx; end
    else begin fa := fy; y := y + dy; end;
    until (x = r) and (y=0);
    end; (Okrąg)
```

Zastosowana tutaj procedura PLOT (X,Y) zapala na ekranie punkt o współrzędnych x,y. Korzystając z symetrii okręgu możemy uprościć algorytm zastosowany w powyższej procedurze. Jeśli ograniczymy się do wyznaczenia punktów okręgu tylko w czwartej ćwiartce układu współrzędnych, to Δx i Δy będą stale równe -1. Mając współrzędne punktu okręgu z czwartej ćwiartki (x,y) i współrzędne środki okręgu (x0,y0) możemy kreślić jednocześnie cztery symetryczne punkty o następujących współrzędnych

1. x0 + x,y0 + y
2. x0 + x,y0-y
3. x0-x,y0+y
4. x0-x,y0-y

Prezentowana poniżej procedura stosuje omówione uproszczenia:

```
procedure Okrag_Nowa_Wersja (x0,y0,r:integer);
var
x,y, fa,fx,fy:integer
begin
  x:=r; y:=0; fa:= 0;
repeat
  Plot(x0 + x,y0 + y); Plot (x0 + x,y0-y);
  Plot(x0-x,y0 + y); Plot (x0-x,y0-y);
  fy : = fa - y shl 1 + 1; fx : = fa - x shl 1 + 1;
  if abs(fx)<abs(fy)
  then begin fa := fx; x := x — 1; end
  else begin fa: = fy; y: = y - 1; end;
until x<0;
end; {Okrąg}
```

Gdybyśmy ograniczyli się tylko do okręgów, rezygnując z elips, można by wykorzystać dodatkową symetrię, kreśląc figurę tylko w połowie pierwszej ćwiartki Dodatkowe cztery punkty powstałyby przy zamianie miejscami współrzędnych x i y w poprzednich czterech punktach. Kolejnym udoskonaleniem algorytmu jest uwzględnienie trzech możliwych do wyboru punktów (dodatkowy ruch, gdy zwiększamy jednocześnie x i y). W tym wypadku trzeba wprowadzić nową zmienną fxy aktualizowaną wg następującego wzoru:

fxy: = fx + fy - fa.

Zmienia się tez kryterium wyboru, ponieważ musimy uwzględnić wartości funkcji w trzech okolicznych punktach. Kolejna procedura zawiera opisane poprawki i pozwala otrzymać okręgi bardziej gładkie niż poprzednio

```
procedure Okrag_Nowa_Wersja_8p (x0,y0,r:integer);
var
  x,y,fa.,fx,fy,fxy,ax,ay,axy:integer
begin
  x := r; y := 0; fa := 0;
repeat
  Plot (x0+x,y0+y); Plot (x0+x,y0-y);
  Plot (x0-x,y0+y); Plot (x0-x,y0-y);
  fx := fa -x shl 1 + 1; ax: = abs(fx);
  fy := fa -y shl 1 + 1; ay: = abs(fy);
  fxy := fx + fy - fa; axy: = abs(fxy);
  if (ax<= ay) and (ax<=axy)
  then begin fa:=fx; x:=x - 1; end
  else if (ay<ax) and (ay<=axy)
  then begin fa:= fy; y:= y --1; end
  else begin fa:= fxy;
            x:=x-1;y:=y-1; end;
  until x<0;
end; (Okrąg)
```

Rezygnując w ostatnich dwóch przykładach procedury okrąg z kreślenia całej figury odeszliśmy trochę od naszego przykładowego plotera, który nie korzysta jednocześnie z czterech piór. Pozbyliśmy się też procedury, która zachowując się poprawnie na całym okręgu, mogłaby służyć do generacji łuków. Natomiast porównując ze sobą dwie ostatnie wersje można stwierdzić, że druga z nich jest bardziej efektywna i daje na ekranie cieńsze linie. Ilość kroków iteracji na każdą ćwiartkę okręgu wynosi 2R dla procedury wybierającej spośród dwóch okolicznych punktów i √2R dla drugiej. Podobnie długości wykreślonych w każdej ćwiartce łuków wynoszą odpowiednio 2R i 4(√2-1)R ≈ 1.657R dla każde| z procedur Ostatnia wartość zbliżona jest do πr/2 ≈ 1.571, co świadczy o tym, że algorytm wybierający spośród trzech okolicznych punktów zapewnia bardzo dobre przybliżenie okręgu

Metody przedstawione w przykładach mogą służyć do wyprowadzenia bardzo efektywnych wzorów interpolacyjnych

do kreślenia krzywych na płaszczyźnie. Jeśli ograniczymy się do krzywych stopnia co najwyżej drugiego: proste, okręgi, elipsy, parabole i hiperbole, to najbardziej skomplikowaną operacją arytmetyczną we wzorach będzie mnożenie dwóch liczb całkowitych. Niepotrzebne są żadne pierwiastki, czy funkcje trygonometryczne. W opisanym ponizej zbiorze GRAPH.SYS (listingi) można znaleźć dalsze przykłady na potwierdzenie tego stwierdzenia.

Zbiór GRAPH.SYS — procedury pascalowe kreślące niektóre krzywe.

Zbiór ten zawiera 5 procedur realizujących podstawowe funkcje graficzne. Wypiszmy ich nagłówki:

1. Procedure PLOT (x,y, Pflag :integer);
2. Procedure LINE (x1,y1. x2,y2. Pflag : integer);
3. Procedure CIRCLE (x0,y0,R,Pflag :integer);
4. Procedure ELLIPSE (x0.y0, a.b, Pflag :integer):
5. Procedure BOX (x1,y1,x2,y2:integer);

Procedura plot zapala lub gasi na ekranie punkt o współrzędnych x,y. Prezentowana wersja działa na komputerze Amstrad PCW 8256. W przypadku korzystania z innej maszyny musimy napisać ją sami. Wszystkie następne procedury są już całkowicie niezależne do sprzętu.

Procedura LINE kreśli linię prostą między punktami o współrzędnych x1, y1 i x2, y2. Procedura CIRCLE wyświetla na ekranie okrąg o środku w punkcie x0,y0 i promieniu R. Jej zawartość przypomina ostatnią wersję przykładowej procedury OKRĄG. Dokładniej biorąc, z powodu różnic w rozdzielczości pionowej i poziomej ekranu, procedura wylicza elipsę o stosunku półosi 1:2. Procedura ELLIPSE kreśli elipsę o środku w punkcie x0,y0 i o półosiach a,b. Z powodu ograniczenia zakresu liczb całkowitych dla 16-bitowego typu INTEGER konieczne jest wykonywanie niektórych rachunków na liczbach rzeczywistych. Spowalnia to nieco obliczenia. Sytuację mogłoby poprawić wprowadzenie w assemblerze arytmetyki 24-bitowych liczb całkowitych. W dotychczas omówionych procedurach zmienna Pflag może przyjmować dwie wartości 0 i 1. Dla jedynki figura jest kreślona na ekranie, a dla zera wymazywana. Ostatnią procedurą pakietu jest procedura BOX rysująca prostokąt, którego lewy, górny róg ma współrzędne x1, y1, a dolny, prawy x2,y2.

Zbiór GRAPH.sys może być dołączany do innego programu przez użycie w tym programie deklaracji typu INCLUDE:

(*$I GRAPH.sys *)

Program demonstracyjny — DEMO.PAS

Listing 2 to zbiór DEMO.pas zawierający pascalowy program demonstrujący możliwości graficzne zbioru GRAPH. SYS. Program ten uruchomiono na komputerze Amstrad PCW 8256 i dalszy sposób postępowania dotyczy właśnie tego sprzętu

Aby otrzymać zbiór wynikowy programu DEMO razem z dołączonym RSX'em musimy wykonać następujące czynności:

1. kompilacja na dysk zbioru DEMO.PAS przy zmniejszonym adresie końcowym zbioru wynikowego. Dołączenie RSX'a (zbiór PLOT.RSX — patrz artykuł w Klanie Amstrada) do systemu operacyjnego obniży górny adres TPA (ang. Transient Program Area), zmniejszając pamięć dostępną dla programu użytkownikowa. Wpisanie C000 Hex w menu wyboru opcji sposobu kompilacji (patrz opis Turbo-Pascal'a) jako adresu końcowego (ang. END ADDRESS) zapewnia duży margines bezpieczeństwa przy zastosowaniu nawet większej ilości RSX'ów.

2. Zbiór DEMO.COM należy poddać działaniu programu systemowego GENCOM.COM:

A>GENCOM DEMO.COM PLOT.RSX <CR> W trakcie tych wszystkich operacji na dysku A: muszą znajdować się następujące zbiory: DEMO.PAS, GRAPH.SYS, TURBO. COM. GENCOM.COM, PLOT.RSX. Wynik działania programu przedstawia rysunek 2. Jednoczesne naciśnięcie klawiszy EXTRA i PTR daje zrzut ekranu na drukarkę

Jarosław Młodzki

Literatura.

1. M. Pietruszka, T. Jedynak. Komputer, 12,87, s. 18-21.
2. R. Wacławek, Komputer 12,87, s. 35-36.
3. P.E. Danielsson, IEEE Trans, Comput, Vol. C-19, Sep. 1970.
4. B.W. Jordan et al., IEEE Trans. Comput., Vol. C-22, Dec. 1973.

LISTING 1

```
(*****************************************************************)
(*                     Zbior GRAPH.SYS                           *)
(*                        ver. 1.01                              *)
(*             (C) Jaroltaw Mlodzki Luty 1988                    *)
(*                                                               *)
(*    Umozliwia realizacje podstawowych funkcji graficznych      *)
(*    na komputerach AMSTRAD PCW 8256/8512 w Turbo-Pascalu       *)
(*    pod kontrola systemu operacyjnego CP/M Plus.               *)
(*****************************************************************)
(*    Lista procedur:                                            *)
(*****************************************************************)
(*    1. PLOT  ( x,y, Pflag ; integer )                          *)
(*    2. LINE  ( x1,y1,x2,y2, Pflag : integer )                  *)
(*    3. CIRCLE  ( x0, y0, R, Pflag : integer )                  *)
(*    4. ELLIPSE ( x0, y0, a,b, Pflag : integer )                *)
(*    5. BOX ( x1,y1,x2,y2 )                                     *)
(*****************************************************************)
(*    Uwagi                                                      *)
(*    1. Do wspolpracy konieczny jest zbior PLOT.RSX             *)
(*       realizujacy dostep do pamieci ekranu                    *)
(*    2. Zmienna Pflag przyjmuje dwie wartosci                   *)
(*       Pflag = 0, wymazuje figure,                             *)
(*             = 1, kresli figure.                               *)
(*****************************************************************)

(**********************************************)
(*  Blok parametrow przekazywanych do wolanego *)
(*  przez procedure PLOT RSX'a                 *)
(**********************************************)
var
    gpb : array(,0,,4,) of integer;

procedure PLOT (x y, PFlag :integer):
(****************************************)
(* procedura PLOT zapala lub gasi punkt        *)
(* o wspolrzednych x,y                   *)
(* Wykorzystuje odwolanie funkcji BDOS    *)
(* o numerze 76 instalowane przez RSX'a   *)
(****************************************)
begin
     gpb(,0,) ; = pFiag;
     gpb(,1,) ; = x;
     gpb(,2,) : = y;
     bdos(76, ader(gpb));
end;  (* of PLOT *)

procedure LINE (x1,y1, x2,y2, PFlag : integer);
(****************************************)
(* procedura kresli linie miedzy para     *)
(* punktow o wpolrzednych x1,y1 a x2, y2,  *)
(* wykorzystuje algorytm 8-punktowy       *)
(****************************************)
var x,y, dx,dy, vy,vy,
    ax,ay,axy,  fa,fx,fy,fxy : integer;
begin
  if x2>x1 then begin dx :=  1;   vx := y2 - y1;  end
          else begin dx := -1;   vx := y1 - y2;  end;
  if y2>y1 then begin dy :=  1;   vy := x1 - x2;  end
          else begin dy := -1;   vy := x2 - x1;  end;
  fa := 0;                x := x1;    y := y1;
  PLOT (x2,y2,PFlag);
  while (x<>x2) or (y<>y2)
  do begin
     plot(x,y,PFlag);
     fx := fa + vx;          ax := abs(fx);
     fy := fa + vy;          ay := abs(fy);
     fxy:= fx + fy - fa;     axy:= abs(fxy);
     if (ax<=ay) and (ax<=axy)
     then begin        fa := fx;    x:= x + dx; end
     else if (ay<ax) and (ay<=axy)
          then begin fa := fy;    y:= y + dy; end
          else begin fa := fxy;   x:= x + dx;
                                  y:= y + dy; end;
     end;       (* of while *)
end; (* of LINE *)

procedure CIRCLE (x0,y0, R, PFlag : integer)
(***********************************************)
(* procedura kresli okrag o srodku w punkcie x0, y0 *)
(* i promieniu R , 2 powodu roznic rozdzielczo ci   *)
(* pionowej i poziomej ekranu, procedura wylicza    *)
(* elipse o stosunku polosi 1:2                     *)
(***********************************************)

var x,y, fa,fx.fy,fxy, ax,ay,axy : integer
begin
  x := r;   y := 0;    fa := 0; repeat
  Plot ( x0+x, y0+y, PFlag);  Plot ( x0+x,  y0-y, Pflag);
  Plot ( x0-x, y0+y, PFlag);  Plot ( x0-x1, x0-y, Pflag);
  fx  := fa - x shl 1 + 1;    ax := abs(fx);
  fy  := fa - y shl 3 + 4;    ay := abs(fxy);
  fxy := fx - fy - fa;      axy:= abs(fxy);
  if (ax<=ay) and (ax<=axy)
  then begin       fa:= fx;     x := x - 1; end
  else if (ay<ax) and (ay<=ay)
     then begin    fa := fy;    y := y-1; end
     else begin    fa := fxy;   x := x-1;
                                y := y - 1; end;
  until   x<0;
end; (* CIRCLE *)

procedure ELLIPSE (x0,y0, a,b, PFlag : integer);
(*********************************)
(* procedura kresli elipse o srodku *)
(* w punkcie x0,y0 i polosiach a,b  *)
(*********************************)
var x,y   : integer;
    fa,fx,fy,fxy, ax,ay,axy, aa,bb, a2,b2 : real; begin
    x  := a;  y  := 0;       fa := 0;
    aa := a;  aa := aa*aa;   a2 := 2*aa;
    bb := b;  bb := bb*bb;   b2 ;= 2*bb;
    repeat
      PLOT ( x0+x, y0+y, PFlag); Plot ( x0+x, y0-y, PFlag);
      PLOT ( x0-x, y0+y, PFlag); PLOT ( x0-x, y0-y, Pflag):
      fx := fa - x * b2 + bb;    ax := abs(fx);
      fy := fa - y * a2 * aa;    ay := abs(fy);
      fxy:= fx + fy - fa;        axy:= abs(fxy);
    if (ax<=ay) and (ax<=axy)
    then begin          fa := fx;    x := x - 1; end
    else if (ay<ax) and (ay<=axy)
         then begin    fa := fy;    y := y - 1; end
         else begin    fa := fxy;   x := x - 1;
                                    y := y - 1; end;
  until   x<0;
end; (* ELLIPSE *)

procedure BOX ( x1,y1, x2, y2 : integer);
(*****************************************)
(* procedura kresli pudelko o wspolrzednych:  *)
(* lewy gorny rog - x1,y1,                    *)
(* prawv dolny rog - x2, y2,                  *)
(*****************************************)
const pflag = 1; begin
      LINE ( x1,y1, x2,y1, PFlag);
      LINE ( x2,y1, x2,y2, PFlag);
      LINE ( x2,y2, x1,y2, PFlag);
      LINE ( x1,y2, x1,y1, PFlag);
end; (* of BOX *)
```

LISTING 2

```
Program GraphDemo;
(************************************)
(* demontracja mozliwosci graficznych *)
(*       pakietu GRAPH.SYS            *)
(*       (C) Jaroslaw Mlodzki         *)
(*           Luty 1988                *)
(************************************)

(*$I graph.sys *)

var
   ch : char;
   i  : integer;

begin (* GraphDemo *)
      clrscr;   write(£27£49£27'f'£27'p');

      Box  (0,248,7 9,0);
      Line (0,125,719,125, 1);
      Line (289,0,289,247, 1);
      Line (479,0,479,247, 1);
      Line (289,60,479,60, 1);
      GotoXY (1,1): for i:=1 to 30 do write(' ');
      GotoXY (34,1); write ('GRAPH DEMO (C) JM 1988');
      GotoXY (13,3): write (' LINE ');
      GotoXY (41,3): write (' ELLIPSE ');
      GotoXY (72,3); write (' LINE ');
      GotoXY (13,18); write(' CIRCLE ');
      GotoXY (18,13); write(' PLOT ');
      GotoXY (72,18); write(' CIRCLE ')
      GotoXY (43,26); write(' BOX ');
      for i:=1 to  12 do Circle (120,62,5+i*5,1);
      for i:=1 to   8 do Eclipse(359,80,80,5+i*5,1);
      for i:=1 to  12 do Circle (599,62, 5+i*5, 1);
      for i:=1 to  25 do Line (120,180,2-Random(238),27-Random(99),1)
      for i:=1 to 500 do Plot (242+Random(235),6+random(50), 1);
      for i:=1 to  25 do Line (599,180,480+Random(288),127+Random(95), 1);

       GotoXY (37,38);  write (' Nacisnij klawisz ');
       rapeat until KeyPressed;
       read(kbd,ch);
       clrscr;    write(£27£49£27'e'£27'q');
end.
```

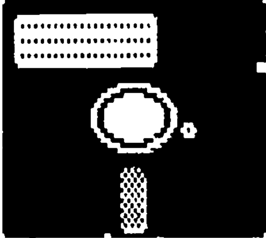
KODY ASCII

32		90	Ź
33	!	91	ź
34	"	92	Ł
35	#	93	ń
36	$	94	ś
37	%	95	_
38	&	96	ą
39	'	97	a
40	(	98	b
41	)	99	c
42	*	100	d
43	+	101	e
44	,	102	f
45	-	103	g
46	.	104	h
47	/	105	i
48	0	106	j
49	1	107	k
50	2	108	l
51	3	109	m
52	4	110	n
53	5	111	o
54	6	112	p
55	7	113	q
56	8	114	r
57	9	115	s
58	:	116	t
59	;	117	u
60	<	118	v
61	=	119	w
62	>	120	x
63	?	121	y
64	@	122	z
65	A	123	ó
66	B	124	ł
67	C	125	ż
68	D	126	ć
69	E	127	
70	F		
71	G		
72	H		
73	I		
74	J		
75	K		
76	L		
77	M		
78	N		
79	O		
80	P		
81	Q		
82	R		
83	S		
84	T		
85	U		
86	V		
87	W		
88	X		
89	Y		

CENY NA DZIEŃ 14.05.1988

	GIEŁDA BAJTKA (tys. zl)	PEWEX BALTONA (USD)	Wielka Brytania (funty)
SINCLAIR			
ZX Spectrum	135	115	50
ZX Spectrum Plus	160	–	60
ZX Spectrum 128 kB + 2	280	–	100
ZX Spectrum 128 kB + 3	350	–	175
Drukarka SEIKOSHA GP 50S	110	–	40
TIMEX 2048	160	146	–
Joystick	6–10	4–12	3–12
COMMODORE			
C–16	90	–	55
C–116	75	–	45
C + 4	150	–	70
C64	235	219	155
C128	360	299	225
C128D	840	–	310
Amiga 500	1000	–	445
Magnetofon 1530, 1531	50	48	25
Stacja dyskietek 1541	275	–	170
ATARI			
65 XE	175	125	65
130 XE	275	199	100
XC 12	50	48	20
Stacja dyskietek 1050	220	185	99
LDW Super 2000	300	199	–
AMSTRAD			
464 z mon. monochromat.	320	–	155
464 z mon. kolor.	405	–	180
6128 z mon. mono.	590	–	230
6128 z mon. kolor.	685	–	310
Stacja dyskietek 3" do 464	300	–	115
PC 1512 SD MM	950	–	300

Sklep Bajtka ul. Koniewa 6 w BYTOMIU tel. 81-57-01	
ZX-81	30.000
ZX-Spectrum 48 KB	110–120.000
ZX-Spectrum plus	150–160.000
SeikoshaGP50	100–110.000
Joystick	7– 14.000
C64	260–270.000
C128	350–400.000
Magnetofon 1531	55– 60.000
Stacja dyskietek 1541	250–260.000
Stacja dyskietek 1571	330–360.000
Drukarka MPS 803	200–250.000
Dyskietki 5.25	1–1. 600
Dyskietki 3.5	3–3. 500
Atari 65 XE	170–180.000
Atari 130 XE	260–270.000
Atari 1050	260–270.000
ATARI 520 ST+SF354	990.000



INDYWIDUALNY BANK DANYCH

Piotr Ziziewicz, uczeń lat 14. Posiada mikrokomputer Spectrum +. Zainteresowania: sport oraz gry komputerowe. Proponuje wymianę gier. Adres: 16-400 Suwałki, ul. Antoniewicza 1a/30.

Norman Patuła, lat 16. Interesuje się informatyką oraz elektroniką. Pragnie nawiązać kontakt z osobami, które posiadają mikrokomputer Texas Instruments. Adres: Poznań, Os. Kwiatowe, ul. Malwowa 20.

Łukasz Curzydło, lat 15. Jest posiadaczem mikrokomputera Commodore C 64 z magnetofonem. Oprogramowanie: gry, programy użytkowe oraz muzyczne. Oczekuje wymiany doświadczeń oraz oprogramowania. Adres: 76-200 Słupsk, ul. Władysława IV 1/29.

Janusz Kasperczyk, uczeń L.O., 18 lat. Posiada CPC 6128. Oprogramowanie: około 60 programów firmowych. Zainteresowania: informatyka, elektronika oraz fotografia. Proponuje wymianę oprogramowania oraz doświadczeń. Adres: 28-400 Pińczów, ul. Bat. Chłopskich 119.

Grzegorz Śliżewski, lat 17, uczeń szkoły średniej. Posiada mikrokomputer AMIGA 2000, monitor 1081, dwie wbudowane stacje dysków 3,5. Pragnie nawiązać kontakt z użytkownikami AMIGI, w celu wymiany doświadczeń, literatury i oprogramowania. Adres: 71 -660 Szczecin, ul. Wilcza 4a.

Dariusz Gorczyński, lat 15. Posiada mikrokomputer ZX Spectrum, około 200 programów. Interesuje się elektroniką oraz informatyką. Zainteresowany jest wymianą programów.

Tadeusz Kisielewicz, Atari 800 XL. Proponuje współpracę i wymianę programów brydżowych. Adres: 77-400 Złotów. Al. Piasta 14/23.

Radosław Mielczarek, posiada mikrokomputer ATARI 800 XL, joystick i magnetofon. Chce nawiązać kontakt z posiadaczami tego mikrokomputera w celu wymiany oprogramowania i doświadczeń. Adres: 87-408 Ciechocin, woj. Toruń.

Maciej Teichert, uczeń LO lat 17. Posiada mikrokomputer Amstrad CPC 6128. Pragnie nawiązać kontakt z innymi posiadaczami tego mikrokomputera. W swoim archiwum posiada gry oraz programy użytkowe. Adres: 62-200 Gniezno, ul. Tulipanowa 3.

Piotr Rowiński, posiada komputer Timex 2048 z monitorem Neptun. Dysponuje dużą ilością programów użytkowych i około 30 grami. Proponuje wymianę programów i doświadczeń związanych z wykorzystaniem komputera w edukacji. Adres: 74-100 Gryfino, ul. Energetyków 36/1.

Remigiusz Grudzień, lat 10. Posiada mikrokomputer ATARI 800XL z magnetofonem oraz joystick. Zainteresowania: motoryzacja, chemia oraz informatyka. Pragnie nawiązać kontakt z rówieśnikami o podobnych zainteresowaniach. Adres: 98-220 Zduńska wola, ul. Obrońców Westerplatte 16.

Krzysztof Dzięcioł, lat 13 pragnie nawiązać kontakt z posiadaczami mikrokomputera Meritum II. Adres: 42-550 Sosnowiec, ul. XXX-Lecia PRL 111a.

Krzysztof Wawrzyniak, lat 14. Posiada mikrokomputery Spectrum + (około 160 programów użytkowych i gier) oraz Amstrad CPC 464 (bardzo mało programów). Lubi łamać gry i szukać nieśmiertelności. Adres: 44-310 Wodzisław, ul Odległa 130.

WSZYSTKO DLA WSZYSTKICH

ATARI

Szeroki wybór oprogramowania na kasetach i dyskietkach.

- co piąty program bezpłatnie
- gwarancja jakości
- rachunki
- katalogi gratis

ATR-SOFTWARE
66-542 Zwierzyn P-1

MICROMAN

Programy na Atari 800 XL, Spectrum 48 KB, Commodore 16/116/+4 na miejscu lub za zaliczeniem pocztowym. Informacje za załączeniem koperty i znaczka pocztowego. 40-181 Katowice, ul. Osikowa 66, tel. 58-51-06.

Zakład Usług Informatycznych

STUDIO NEXT

poleca:
PROGRAMY DO ATARI

- ceny konkurencyjne do giełdy
- instrukcje
- Bajtkowa lista Przebojów

42-200 Częstochowa
Katedralna 8 m 11
tel.477-71

WOJEWÓDZKIE PRZEDSIĘBIORSTWO HANDLU WEWNĘTRZNEGO ODDZIAŁ W TYCHACH

VIDEOBIT

43-100 Tychy, aleja ZMP 77
tel. 27-69-75

- **minikomputery 8-bitowe (Atari, Commodore, Schneider-Amstrad)**
- **minikomputery 16-bitowe kompatybilne z IBM PC**
- **drukarki 10" i 15" firm STAR, EPSON, AMSTRAD**
- **magnetowidy**
- **kamery video**
- **anteny satelitarne**
- **aparaturę badawczo-naukową**

Zapewniamy o atrakcyjnych cenach.

ATARI ZX SPECTRUM, TIMEX

programy, instrukcje, nowości oferuje:

SPEKTRA

21-422 Stanin
Informacje kopertą zwrotną.

COMMODORE 16,116,4/PLUS ATARI XE, XL, ST, SHARP

Studio komputerowe

„CANON"

Proponuje bogatą ofertę oprogramowania.
Koperta zwrotna
41-506 Chorzów
ul. Karłowicza 23/12

Najnowsze oprogramowanie

ATARI ● AMSTRAD
COMMODORE
SPECTRUM

oferuje

„ATARES"

— spółka z o.o. Chorzów, Jesionowa 3, tel. 417-573 g. 9.00 — 17.00. Sprzedaż wysyłkowa. Dla rzemiosła i przedsiębiorstw oprogramowanie „pod użytkownika". Kupno-sprzedaż sprzętu komputerowego i audiowizualnego — niskie ceny. Dla instytucji rachunki.

JOYSTICK SERVICE

****   naprawa joysticków***
***** wymiana standardowych styków na mikroprzełączniki***

Warszawa tel 41-22-22
(przyjmujemy przesyłki polecone)

Gry, programy użytkowe, opisy na Atari XL/XE oferuje „MIKROFAN", 45-064 Opole 1, skr. poczt. 158. (informacje za załączeniem znaczka).

STUDIO KIJOWIANKA
AMSTRAD ATARI XL, XE, ST
COMMODORE 64, 128
Poleca literaturę i programy na kasetach i dyskach.
Warszawa, ul. Targowa 26. Rachunki oraz wysyłka pocztą.
Informacje za załączeniem koperty i znaczka.

Klawiatury Commodore 610 sprzedam. Białystok 75-35-31.

SPRZĘŻENIE ZWROTNE

Drogi Bajtku!

W odpowiedzi na list czytelnika dotyczący WARSAW BASIC-a (Bajtek nr 2/88 str. 29) stwierdza Pan, że polski kompilator Basic-a sprzedawany jest bez instrukcji.

W załączeniu do tego listu przesyłam jeden egzemplarz instrukcji sprzedawanej razem z kasetą, na której nagrany jest kompilator TOBOS-FP. Zdaniem użytkowników 9 stron instrukcji obsługi jest w tym wypadku zupełnie wystarczającą liczbą.

Naszym celem było stworzenie efektywnego urządzenia, którego używanie nie wymaga studiowania tomów instrukcji i myślę, że to się udało.

Wojciech Skaba
ul. Bożyńskich 38 m. 23
87-100 Toruń

Dziękuję za list i za instrukcję — faktycznie zwięzłą i dość wyczerpującą, o ile zdołałem się zorientować. Moja wypowiedź w numerze 2/88 nie miała bynajmniej charakteru zarzutu pod adresem któregokolwiek z wydanych programów, lecz raczej sygnalizacji pewnego problemu o charakterze ogólnym. Mianowicie działalność wydawnicza na rynku programów nie może na dłuższą metę sprowadzić się tylko do nagrywania kaset (czy dyskietek).

Nie przeczę, że TOBOS-FP może być wystarczająco opisany przy pomocy tylko 9 stron tekstu. Taką sytuację trzeba jednak, przyzna Pan, traktować jako szczęśliwy wyjątek. TOBOS-FP korzysta w maksymalnym stopniu z oprogramowania systemowego ZX Spectrum — edytora, dialektu Basic-a. Oznacza to, że lwią część podręcznika użytkownik TO- BOSA-FP już otrzymał wcześniej w postaci podręcznika obsługi komputera. W przeciwnym przypadku nie obeszłoby się prawdopodobnie bez wydania instrukcji, w formie książki. Tak też jest w przypadku wszystkich niemalże znanych mi komplitatorów zagranicznych. W przypadku translatorów bardziej wyszukanych podstawowy podręcznik użytkownika liczy bardzo często 400 — 600 stron.

Mam prośbę związaną z programowaniem w języku Turbo Pascal. Nigdzie nie mogę znaleźć instrukcji, która byłaby odpowiednikiem INKEY$ w Basic-u. Jeżeli to możliwe, proszę o podanie tej instrukcji i jej składni.

Adam Czajka
Os. Rusa 103/6
61-245 Poznań

MW celu pobierania informacji o stanie klawiatury używa się standardowej procedury **read**, przy czym czytanie powinno odbywać się z pliku powiązanego z urządzeniem **KBD**:. Standardowym takim plikiem jest **Kbd**. Tak więc w czytaniu z klawiatury znaku i przypisaniu go na zmienną Ch typu char odpowiada wywołanie

read (Kbd, Ch);

Dodatkowym narzędziem jest standardowa funkcja, której deklaracja można przedstawić jako

function KeyPressed: boolean;

Jej wynik wskazuje, czy w momencie wywołania jest wciśnięty jakikolwiek klawisz.

Tak więc w większości przypadków, gdy nie chodzi nam np. o jednoczesne działanie programu i kontrolowanie stanu klawiatury, w celu pobrania znaku będziemy używać konstrukcji

repeat until KeyPressed;
read (Kdb, Ch);

Z urządzeniem KBD: możemy też powiązać dowolną zmienną typu text (file of char) poprzez użycie

Assign „F, 'KBD:');

Opisane konstrukcje mają zastosowanie w wersji Turbo Pascal 3.0 i wcześniejszych. W wersji 4,0 zastosowano całkiem inne rozwiązanie.

Jestem posiadaczem ATARI 800XL i stacji dysków 1050. Planuję rozszerzyć ten zestaw o drukarkę STAR LC-10. Słyszałem, że jest to nowy produkt firmy, dlatego też chciałbym dowiedzieć się, jakie są opinie o tej drukarce. Przede wszystkim interesuje mnie druk polskich liter w trybie

AGENCJA INFORMATYCZNA "Beta B"
Telef. 69-03-85
SKRYTKA P-254
41-200 SOSNOWIEC
oferuje, również wysyłkowo-pocztą:
PROGRAMY,
INSTRUKCJE,
OPISY I SCHEMATY TECHNICZNYCH UDOSKONALEŃ KOMPUTERÓW
ACORN AMSTRAD ATARI
COMMODORE IBM SHARP
KATALOGI INFORMACYJNE-BEZPŁATNE

NLQ. Czy byłoby możliwe zastąpienie jednego zestawu znaków międzynarodowych? Jeśli nie, bardzo prosiłbym o wskazanie mi programu realizującego powyższe zadanie.

Dariusz Matuszewski (adres do wiadomości redakcji)

Test drukarki LC10 zamieściliśmy w „Bajtku" „Tylko o Commodore". Przedstawiony tam model różni się od standardowego tylko zamontowanym na stałe interface'em do Commodore.

Zestawy znaków są przechowywane w pamięci ROM drukarki. Możliwe jest zdefiniowanie polskich znaków poprzez wstawienie własnego układu (układów) EPROM w miejsce układów firmowych. Rozwiązania programowe tego problemu z reguły sprowadzają się do druku polskich liter w trybie graficznym, co wymaga niebanalnego programu sterującego wydrukiem i znacznie spowalnia tempo pracy drukarki. Redakcja nie posiada informacji o takim programie dla zestawu ATARI XL—LC-10.

Chciałbym nawiązać do cyklu artykułów opisujących Hisoft Pascal. Otóż w pierwszej części tego cyklu (nr 9/87 str. 11) został opisany program PRINT 64 współpracujący z Hisoft Pascal-em HP4S. Program ten może współpracować też z wersją HP4TM16. Nie to jest jednak najważniejsze. Do napisania tego listu skłonił mnie fragment artykułu opisujący powrót do 32-kolumnowej organizacji ekranu (przez powrót do Basic-a zleceniem B, wykonanie NEW i ponowne uruchomienie Pascal-a przez RANDOMIZE USR 24603). Nie został natomiast opisany inny sposób, znacznie wygodniejszy i nie powodujący skutków ubocznych, takich jak skasowanie programu ładującego i utrata możliwości nagrania programu na taśmę oraz brak możliwości powrotu do trybu 64-kolumnowego. Sposób ten jest następujący: jako komendę edytora podajemy CAPS SHIFT + 3. Potem w każdej chwili możemy wrócić do 64 kolumn ponownie wciskając te same klawisze. Mam nadzieję, że ta uwaga przyda się tym czytelnikom, którzy nie odkryli jeszcze tej dodatkowej możliwości.

Jan Kondraciuk ul. Słowackiego 25/60 05-120 Legnica

Faktycznie opisu podanej funkcji edytora zabrakło w cyklu „Bez wyboru". Autor opierał się w tym miejscu na podręczniku nieco starszej wersji kompilatora Hisoft Pascal, stąd też brak owej wzmianki. Podany sposób pozostaje jednak nadal skuteczny, jeżeli nam zależy na fizycznym usunięciu z pamięci wszystkiego poza samym kompilatorem. Może się to zdarzyć w przypadku uruchomienia bardziej rozległych programów.

W numerze 3/88 „Bajtka" zamieszczono fotografię przerobionej klawiatury do komputera ZX Spectrum +. Bardzo mnie ten temat zainteresował i bardzo bym prosił, wręcz nalegał, by redakcja udostępniła mi materiały, jak zrobić taką klawiaturę.

Andrzej Dunajec ul. Czerwonych Sztandarów 94a/343 41-303 Dąbrowa Górnicza

Listów tego typu nadeszło ostatnio sporo, dlatego też z braku innej możliwości odpowiedzi odsyłam do literatury. Opis podłączenia dodatkowej klawiatury do ZX Spectrum wraz z potrzebnymi schematami został zamieszczony w numerze 1/1987 Magazynu Komputerowego „Młodego Technika" INFORMIK, w artykule Grzegorza Zalota „Jak z ZX Spectrum zrobić komputer" na str. 18.

Wprawdzie prezentowane tam rozwiązanie różni się od prezentowanego na zdjęciu w „Bajtku", ale jest również dość wygodne (klawiaturę podłącza się z zewnątrz przez złącze krawędziowe, możliwe jest użycie klawiatury z wydzielonym blokiem numerycznym i funkcyjnym itd.).

Jestem użytkownikiem TIMEX-a 2048. W jaki sposób można regulować prędkość zapisu i odczytu programów na taśmie, czy można to regulować przy pomocy jakiejś zmiennej systemowej? Ukazywały się już w „Bajtku" programy tego typu dla Commodore i Atari.

Krzysztof Murawski Zadobrze 12, 18-230 Ciechanowiec woj. łomżyńskie

ZX Spectrum i Timex dysponują dość wysoką prędkością transmisji danych z magnetofonu — na przykład w porównaniu ze standardowymi prędkościami transmisji Commodore i Atari. Przyspieszenie transmisji jest możliwe teoretycznie na drodze czysto programowej, poprzez zastąpienie systemowych podprogramów obsługi transmisji własnymi ich substytutami.

Wiąże się to jednak z dużymi trudnościami natury technicznej. Mianowicie zagęszczenie informacji na nośniku wiąże się oczywiście ze zwiększeniem podatności zapisu i odczytu na błędy i przekłamania. Z tymi ostatnimi i tak nie jest najlepiej ze względu na to, że Spectrum używa zwykłego magnetofonu i — co za tym idzie — komputer analizuje bezpośrednio sygnał akustyczny.

Z tych przyczyn istniejące sposoby przyspieszenia transmisji danych w ZX Spectrum opierają się najczęściej na rozwiązaniach sprzętowo-programowych pozwalających na zminimalizowanie ilości błędów transmisji.

Niemniej nie da się w pełni wykluczyć możliwości programowego rozwiązania problemu, jakkolwiek redakcja „Bajtka" jeszcze z takowym się nie spotkała. W przypadku poznania metody, chętnie się nią z czytelnikami podzielimy.

Marcin Waligórski



PRZYSTAWKA „POLBASIC"

Przystawka POLBASIC nie różni się wielkością od zwykłego interfejsu Kempstona.

Składa się ona z szyny łączącej, wciskanej w gniazdo komputera, z przelotem na drugą stronę, płytki z pamięcią EPROM i kilkoma częściami elektronicznymi. Wszystko obudowane jest pudełkiem z czarnego plastiku z nazwą przystawki. Na zewnątrz wyprowadzone są dwa gniazda typu „jack", oznaczone jako Wejście i Wyjście, dwie diody elektroluminescencyjne podpisane MOTOR i ROM oraz przycisk podpisany NMI.

Po podłączeniu przystawki do komputera i włączeniu go do sieci słychać krótki dźwięk i na dole ekranu ukazuje się napis „TACT POLBASIC" wraz z dwoma różnokolorowymi paskami. Ekran jest czarny, a litery białe w celu oszczędzania oczu i monitora, świeci się też zielona dioda na przystawce.

Po wciśnięciu dowolnego klawisza w oczy rzuca się jedno — wszystkie komendy są po polsku! Na przykład PISZ zamiast PRINT, CZYŚĆ zamiast CLS, czy PRZYWRÓĆ zamiast RESTORE. Niektóre tłumaczenia są wręcz śmieszne. Autor kierował się tym, by tłumaczenia układały się w tekście programu w poprawne polskie zdania. Uważam jednak, i nie tylko ja, że takie tłumaczenie komend na język polski pozbawione jest sensu. Językiem programowania jest język angielski i zmiana na inny będzie zawsze sztuczna i bezcelowa. Tylko LOGO powinno być w rodzimym języku użytkownika. Sensowna jest natomiast myśl zastosowania polskich komunikatów o błędach.

To dopiero początek możliwości przystawki. Przede wszystkim mamy nowy zestaw znaków z pogrubionymi literami i dodatkowymi znakami w miejsce nie używanych kodów. Wyrażenie PI zastąpiono symbolem matematycznym, a SQR — znakiem pierwiastka. Obecne są polskie litery, zamiennie ze znakami graficznymi.

Większość błędów zawartych w ROM-ie Spectrum została poprawiona. Mamy też szereg bardzo przydatnych funkcji. CLS wprowadzone przez CAPS SHIFT i ENTER powoduje, oprócz wyczyszczenia ekranu, zmianę kolorów na przeciwne. Ten sam efekt daje komenda PRZYWRÓĆ 1. PRZYWRÓĆ z kolejnymi cyframi powoduje wywoływanie większości innych funkcji. Wszystkie wywołania opisane są wyczerpująco w instrukcji.

Pożyteczne jest też dodatkowe działanie klawisza CAPS SHIFT. Przytrzymanie go pod koniec ładowania programu z taśmy da ten sam efekt, co CAPS SHIFT i NMI, lecz bez zmiany zawartości rejestrów.

Testowany w redakcji egzemplarz czasem zawieszał się nieodwracalnie lub powodował dziwne zjawiska. Lampka „Motor" lubiła „wariować" i w większości gier powrót do systemu był niemożliwy. Mimo to, podczas blisko miesięcznego intensywnego użytkowania przystawka spełnia swoją rolę.

Jak podaje dystrybutor, na życzenie klientów, opracowana została nowa wersja przystawki o nazwie BASIC PLUS. Posiada ona wszystkie funkcje starej przystawki z niezmienionym angielskim Basic-em i dodatkowo wzbogacona jest o funkcje kasowania pamięci, wyświetlania ilości wolnej pamięci i katalogowania taśmy.

Sumując: przystawka TACT POLBASIC jest użytecznym narzędziem dla każdego, kto chce wzbogacić swój komputer o szereg dodatkowych funkcji, działa z każdym „klonem" ZX Spectrum. Polecam ją nie tylko entuzjastom programowania w Basicu i asemblerze, lecz i graczom. Przystawka kosztowała ostatnio 39000. Myślę, że jest ona warta swojej ceny.

Przystawkę „POLBASIC" otrzymaliśmy do testowania dzięki uprzejmości pana Wojciecha Szantera, dyrektora Centralnej Składnicy Harcerskiej.

Przystawka została zaprojektowana przez p. Macieja Szewczuka ze spółdzielni TACT, a produkowana jest przez przedsiębiorstwo ATUT — spółka z o.o. Rozprowadza ją Centralna Składnica Harcerska poprzez placówki sprzedaży sprzętu komputerowego w całym kraju.

Marcin Przasnyski

KTO bliżej?

Cześć Maluchy!

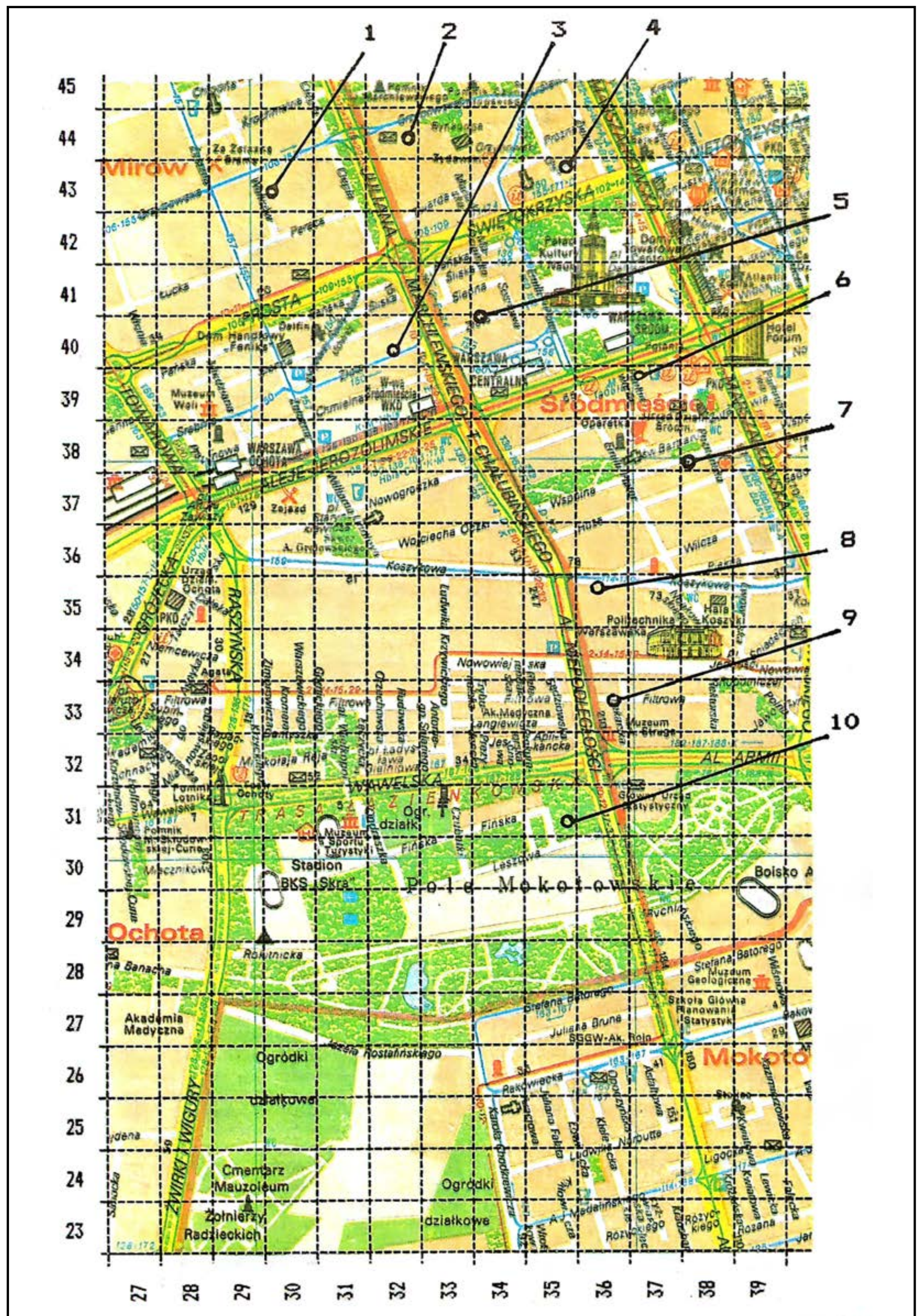
Nie ma chyba rodziny programów bardziej licznej niż bazy danych. Nic w tym zresztą dziwnego. Komputer to narzędzie informatyki, informatyka zajmuje się informacją a informację gromadzimy w bazach danych.

Uważni czytelnicy rubryki „Tylko dla przedszkolaków" pamiętają prawdopodobnie, że już kiedyś konstruowaliśmy razem bazę danych ("Bajtek" nr 9/86). Nazywała się ona „Notesik" i służyła do zapisywania imion, nazwisk i telefonów znajomych. Potrafiła znaleźć numer telefonu podanego kolegi oraz ustawić wszystkie nazwiska w kolejności alfabetycznej. I to wszystko, co mogła zrobić nasza baza danych. Pod względem możliwości niewiele różniła się od zwykłego notesu.

Bywają jednak zupełnie inne bazy danych. Takie, które prócz tego, że przechowują potrzebne nam informacje, potrafią również podejmować za nas pewne decyzje. Zdarza się czasem, że są to decyzje bardzo złożone.

W numerze 5/88 „Bajtka" wspólnie z koleżankami i kolegami założyliśmy własną gazetę. Taki podwórkowy dziennik. Chcielibyśmy oczywiście, by w każdym miejscu, gdzie dzieje się coś ciekawego znalazł się nasz reporter. Często powinien on znaleźć się tam w ciągu kilku minut, a więc powinien to być ktoś, kto mieszka najbliżej.

Studiowanie listy adresów i błądzenie palcem po mapie nie jest najlepszym wyjściem. Może się bowiem okazać, że nim zdecydujemy, komu zlecić napisanie reportażu, nie będzie już o czym pisać. Jedyny sposób na szybkie i bezbłędne wytypowanie odpowiedniego kandydata, to zaprzęgnięcie do tej pracy komputera.




```
100 PRINT "Podaj wsprzdne interesujacego
nas kwadratu."
110 PRINT "Pozioma:"
120 INPUT x
130 PRINT "Pionowa:"
140 INPUT y
150 LET l=1000
200 READ a$
210 IF a$="koniec" THEN GOTO 250
220 READ nr,x1,y1
230 IF SQR((x-x1)^2+(y-y1)-2)<l THEN LET
l=SQR((x-x1)^2+(y-yl)^2): i$=a$: n=nr
240 GOTO 200
250 PRINT "Najblizej bedzie mial(a)
";i$;" (nr:";n;")"
1010 DATA "Barnaba",1,30,43
1020 DATA "Maly Kazio",2,32,44
1030 DATA "Zdzich Marchewka",3,32,40
1040 DATA "Asia",4,35,43
1050 DATA "Jacek",5,34,40
1060 DATA "Marysia",6,37,39
1070 DATA "Kubus Literka",7,38,38
1080 DATA "Marek i Wladek",8,36,35
1090 DATA "Zosia",9,36,33
1100 DATA "Krysia",10,35,31
1110 DATA "koniec"
```

Za każdym razem, gdy przyjdzie nam ochota zasto sować komputer do zadań praktycznych, musimy się zastanowić, w jakiej postaci zapisać wszystkie dane, aby były one zrozumiałe dla komputera. Jeśli na przykład podamy komputerowi informację, że Kubuś mieszka przy ul. Wspólnej, a Zdzich Marchewka przy Złotej, to komputer nie będzie w stanie stwierdzić, czy jest to daleko czy blisko. Oczywiście, można próbować „nauczyć" go planu Warszawy. Jest to jednak zajęcie bardzo żmudne i przekraczające z całą pewnością umiejętności komputerowych przedszkolaków. My poradzimy sobie w nieco inny, o wiele bardziej prosty sposób. Podzielimy plan miasta (oczywiście może to być wasze rodzinne miasto) na małe kwadraty, które następnie ponumerujemy w poziomie i pionie tak jak na naszym rysunku. Tak więc zamiast podawać adres w postaci nazwy ulicy, numeru domu i mieszkania, wprowadzimy do komputera jedynie informację w którym kwadracie znajduje się dom danej osoby.

Odległość pomiędzy poszczególnymi kwadratami obliczymy bardzo łatwo stosując twierdzenie Pitagorasa. No proszę, kto powie jak ono brzmi?

Jeżeli trójkąt jest prostokątny, to suma pól kwadratów zbudowanych na jego przyprostokąt- nych, jest równa polu kwadratu zbudowanego na przeciwprostokątnej. Wynika stąd, że odległość dwóch punktów jest równa pierwiastkowi z sumy kwadratów odległości poziomej i pionowej:

$$1 = \sqrt{x^2 + y^2}$$

Jest to oczywiście odległość w linii prostej i w związku z tym komputer nie bierze pod uwagę ukształtowania ulic czy jakichś przeszkód terenowych. Może się zdarzyć, że prawdziwa droga będzie znacznie dłuższa od teoretycznie obliczonej.

Nasz program skonstruowany jest w taki sposób, że komputer pyta nas o współrzędne (numery) interesującego nas kwadratu. Po ich wprowadzeniu odczytuje kolejno informacje o poszczególnych redaktorach i za każdym razem oblicza odległość ich domów od naszego kwadratu. W zmiennych **i$** oraz **n** przechowuje nazwisko i numer tego, który mieszka najbliżej. Po sprawdzeniu wszystkich, komputer melduje nam, kogo należy wysłać.

A może Wy macie pomysł na podobny program „myślącej bazy danych" przydatny na przykład w szkole. Jeśli tak, napiszcie. Najciekawsze programy wydrukujemy!

Romek

1. *Operator w centrali poszukuje ciężarówki, sygnał przesyłany jest z centrali przewozów linią telefoniczną do komputerowej stacji naziemnej skąd przekazywany jest do odbiornika w ciężarówce via satelita.*

2. *Radioodbiornik w ciężarówce odpowiada wysyłając zakodowany sygnał. Satelity namierzają go ustalając dokładne położenie ciężarówki.*

3. *Informacja przekazywana jest z satelity z powrotem do komputerowej stacji naziemnej, gdzie wyświetlana jest w postaci punktu na mapie generowanej na ekranie monitora. Tą drogą można przesyłać również informacje do i od kierowcy.*

ILLUSTRATION BY HAROLD SMEICER - WSW&WF

Gdzie jest ciężarówka 54?

Złodzieje, którzy na początku tego roku ukradli niedaleko San Bernardino w Kaliforni (USA) ciężarówkę z ładunkiem za 55 tysięcy dolarów, nie mieli szczęścia. W ciągu dwu i pół dnia, kiedy uciekali nią dokoła Południowej Kaliforni. operatorzy w Centrali Krajowych Usług Przewozowych obserwowali każdy ich ruch na ekranach swoich komputerów osobistych.

Skradziona ciężarówka wyposażona była bowiem w nadajnik, którego sygnały odbierane były przez satelitę. Wkrótce policja aresztowała zdumionych złodziejaszków, a kierowcy firmy szybko odnaleźli ciężarówkę zaparkowaną na przedmieściach Los Angeles.

"Błądzącą" ciężarówkę zlokalizowano za pomocą systemu firmy Geostar Corporation z Waszyngtonu testowanego od maja ub. roku przez kilkanaście przedsiębiorstw. Firma jeszcze korzysta z usług francusko-amerykańskiego satelity meteorologicznego, ale wkrótce zaplanowano wyniesienie na orbitę pierwszego z trzech własnych satelitów, co znacznie przyspieszy eksperyment i zamieni go z lokalnego na ogólnokrajowy. I chociaż interes ten jest jeszcze w stadium organizacji, to konkurencja nie śpi: następna firma — Omnimet Corporation z Los Angeles proponuje swoje usługi w tym samym zakresie już od czerwca br.

Założyciele obu firm twierdzą, iż już wkrótce będą mogli odpowiedzieć na pytanie dręczące od lat dyspozytorów dalekobieżnych przewozów: gdzie są moje ciężarówki?

Dzisiaj przejeżdżają one z hukiem międzystanowymi autostradami będąc przez kilkanaście (-dziesiąt) godzin poza zasięgiem dyspozytora, który w razie spóźnienia sygnalizowanego przez odbiorcę nie wie, czy ciężarówka utknęła w korku, zboczyła z trasy czy stoi gdzieś zepsuta, a kierowca szuka telefonu, żeby zawiadomić centralę i wezwać pomoc. Trwa to czasem ponad pół godziny a stracony czas kosztuje.

Ta ciągła niepewność operatorów odpowiedzialnych w centrali za terminowość przewozów wzmaga ich wysiłki w celu zainstalowania nadajników w ciężarówkach.

Zasada współpracy radionadajnika firmy Geostar z satelitą i komputerem w centrali jest następująca: sygnał emitowany z satelity „wyławia" kod radionadajnika poszukiwanej przez centralę ciężarówki i wraca z powrotem na orbitę. Lokalizacja (z dokł. do 1 mili) odbywa się przy udziale przystosowanego do tego eksperymentu systemu nawigacji LORAN-C — z powodzeniem wykorzystywanego do lat przez statki i prywatne samoloty.

W 1990 roku, kiedy Geostar umieści na orbicie drugiego satelitę, pozycja poszukiwanej ciężarówki znana będzie z dokładnością do 10 metrów. Informacje do i z centralnego komputera Geostaru przekazywane będą odbiorcom z poszczególnych linii przewozowych telefonicznie. Operator w centrali będzie mógł w każdej chwili zobaczyć swoją ciężarówkę rzucając okiem na mapę wyświetloną na ekranie osobistego IBM czy APPLE. Ponadto — korzystając z pamięcią komputera Geostaru operator z łatwością odtworzy trasę ciężarówki z kilku poprzednich dni. W razie zmian w organizacji przeładunków będzie mógł zaprogramować również jej nową trasę.

— System ten zrewolucjonizuje naszą pracę — twierdzi Tom Brooke, prezes Frederick Transport, którego już 13 z 620 ciężarówek, kursujących między USA a Kanadą, wyposażonych jest w „elektronicznego pasterza" przez Geostar. Znikną również — kontynuuje Tom Brooke — pretensje fabryk zaopatrywanych przez nasze ciężarówki w systemie „just in time" oraz żądania dokładnego określenia przybycia kolejnej ciężarówki. Oni i my będziemy znali rozkład jazdy.

Mniejszym ryzykiem obarczone będą przewozy ładunków niebezpiecznych i szczególnie wartościowych. Kierowca w przypadku kłopotów jednym naciśnięciem guzika będzie mógł wezwać pomoc określając precyzyjnie swoje położenie.

Z drugiej strony możliwość ścisłej kontroli ma prócz dobrych i złe (dla niektórych kierowców) strony.

Kiedy system Geostar w ramach eksperymentu zainstalowano na ciężarówkach firmy Three Coast Carriers, to pierwsza ciężarówka, którą prezes Riley Crosby zobaczył na ekranie komputera, była oddalona 300 mil od zaplanowanej przez firmę trasy.

Stąd też kierowcy ciężarówek mają mieszane uczucia co do zalet radionadajnika proponowanego przez Geostar. Jeden z nich — Gustaw Gyllenikof z firmy ATA jest niezadowolony z tego, że jak twierdzi, „Wielki Brat" wciąż go obserwuje. Kierowcy zdają sobie sprawę, że będą podlegać jeszcze dokładniejszej kontroli, gdy w latach 90-tych zostanie wprowadzona kieszonkowa wersja radionadajnika.

Według obliczeń prezesa Brooksa, zarówno firma jak i kierowcy wkrótce zyskają na wprowadzeniu tego dość kosztownego systemu — abonament miesięczny radionadajnika na jednej ciężarówce wynosi 165 dolarów.

Około 1/5 tej sumy firma potrącać będzie z pensji kierowcy. Za to — argumentują twórcy systemu — nie będzie musiał on zbaczać z trasy, żeby znaleźć najbliższy telefon, z którego zadzwoni do centrali. Amerykanie wyliczyli, że takie średnie „zboczenie" wynosi 30 mil; obliczyli też, że koszt jednej mili wynosi 80 centów. Wyliczenie jest już zatem proste: każdy telefon do centrali kosztuje extra 24 dolary. Wszystkie telefony w ciągu jednego dnia „to już grube tysiące dolarów — twierdzi prezes Brooks. Kto nie zakupi tego systemu daleko nie zajedzie".

Franciszek Penczek